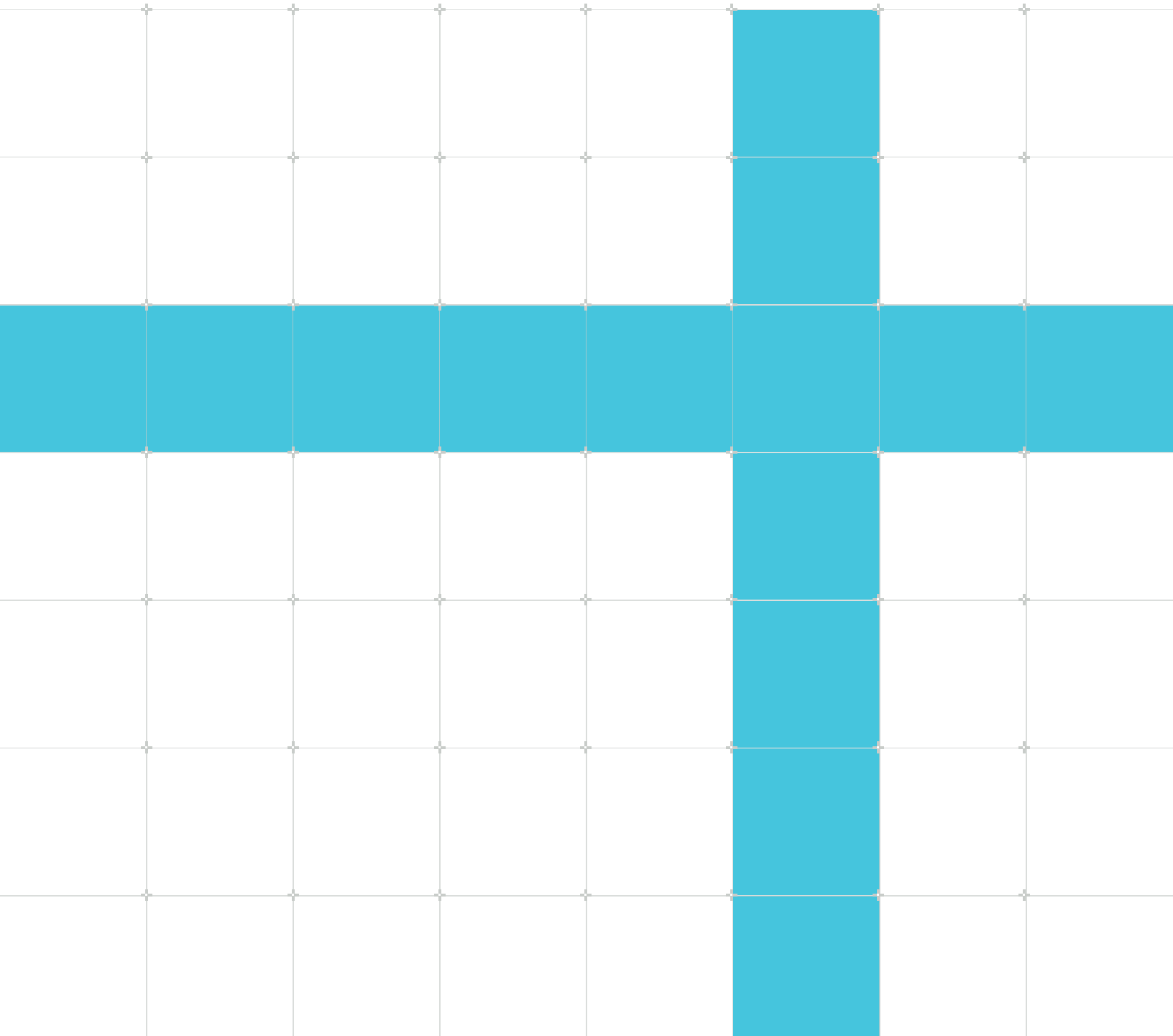
Fast Models

Version 11.16

Fixed Virtual Platforms (FVP) Reference Guide

Issue 00
100966_1116_00_en



Fast Models

Fixed Virtual Platforms (FVP) Reference Guide

Release Information

Document history

Issue	Date	Confidentiality	Change
A	31 May 2014	Non-Confidential	New document for Fast Models v9.0, from DUI0575H for v8.3.
B	30 November 2014	Non-Confidential	Update for v9.1.
C	28 February 2015	Non-Confidential	Update for v9.2.
D	31 May 2015	Non-Confidential	Update for v9.3.
E	31 August 2015	Non-Confidential	Update for v9.4.
F	30 November 2015	Non-Confidential	Update for v9.5.
G	29 February 2016	Non-Confidential	Update for v9.6.
H	31 May 2016	Non-Confidential	Update for v10.0.
I	31 August 2016	Non-Confidential	Update for v10.1.
J	11 November 2016	Non-Confidential	Update for v10.2.
K	17 February 2017	Non-Confidential	Update for v10.3.
1100-00	31 May 2017	Non-Confidential	Update for v11.0. Document numbering scheme has changed.
1101-00	31 August 2017	Non-Confidential	Update for v11.1.
1102-00	17 November 2017	Non-Confidential	Update for v11.2.
1103-00	23 February 2018	Non-Confidential	Update for v11.3.
1104-00	22 June 2018	Non-Confidential	Update for v11.4.
1104-01	17 August 2018	Non-Confidential	Update for v11.4.2.
1105-00	23 November 2018	Non-Confidential	Update for v11.5.
1106-00	26 February 2019	Non-Confidential	Update for v11.6.
1107-00	17 May 2019	Non-Confidential	Update for v11.7.
1108-00	5 September 2019	Non-Confidential	Update for v11.8.
1108-01	3 October 2019	Non-Confidential	Update for v11.8.1.
1109-00	28 November 2019	Non-Confidential	Update for v11.9.
1110-00	12 March 2020	Non-Confidential	Update for v11.10.
1111-00	9 June 2020	Non-Confidential	Update for v11.11.
1112-00	22 September 2020	Non-Confidential	Update for v11.12.

Issue	Date	Confidentiality	Change
1113-00	9 December 2020	Non-Confidential	Update for v11.13.
1114-00	17 March 2021	Non-Confidential	Update for v11.14.
1114-01	14 April 2021	Non-Confidential	Update for FVP_Base_AEMv8R.
1115-00	29 June 2021	Non-Confidential	Update for v11.15.
1116-00	6 October 2021	Non-Confidential	Update for v11.16.

Proprietary Notice

signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Web Address

developer.arm.com

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

To report offensive language in this document, email terms@arm.com.

Contents

1 Introduction

1.1 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm® Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Convention	Use
italic	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace bold`	Denotes language keywords when used outside example code.
`monospace underline`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.
	This represents a recommendation which, if not followed, might lead to system failure or damage.
	This represents a requirement for the system that, if not followed, might result in system failure or damage.
	This represents a requirement for the system that, if not followed, will result in system failure or damage.
	This represents an important piece of information that needs your attention.
	This represents a useful tip that might make it easier, better or faster to perform a task.

Convention	Use
Remember	This is a reminder of something important that relates to the information you are reading.

1.2 Feedback

Arm welcomes feedback on this product and its documentation.

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

Information about how to give feedback on the content.

If you have comments on content then send an e-mail to errata@arm.com. Give:

- The title Fast Models Fixed Virtual Platforms (FVP) Reference Guide.
- The number 100966_1116_00_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Note

Arm tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

1.3 Other information

See the Arm® website for other relevant information.

- Arm® Developer.
- Arm® Documentation.
- Technical Support.
- Arm® Glossary.

2 Introduction to FVPs

Fixed Virtual Platforms (FVPs) enable software development without the need for real hardware.

They are available for Linux and Windows hosts, either:

- As pre-built executables. Their composition is fixed, but you can configure their behavior using parameters.
- As source code examples in the Fast Models product, with the tools required to customize and build them.

FVPs are available for all Cortex®-A, Cortex®-R, and Cortex®-M processors, and most of them support the CADI, MTI, and Iris interfaces, so can be used for debugging and for trace output.

Arm provides validated Linux and Android deliverables for various platforms, including FVPs. For details, see the Arm Development Platforms wiki on Arm Community. To get started with Linux on FVPs, see FVPs on Arm Community.

2.1 Types of FVP

Most Fast Models FVPs are provided in a single license-managed library. In addition, some Architecture Envelope Model (AEM) FVPs are available for download without requiring a license.

Fast Models FVPs are based on the following platforms:

- Base Platform.
- BaseR Platform.
- Arm® Versatile™ Express development boards.
- Arm® MPS2 or Arm® MPS2+ platforms, for Cortex®-M series processors.

FVPs can be obtained in the following ways:

- License-managed FVPs, including some plug-ins and utilities, are provided in the FVP Standard Library. It is available for Windows or Linux hosts. For information on how to download it, see Fixed Virtual Platforms on Arm Developer.
- Free-of-charge AEM FVPs can be downloaded from Arm Architecture Models on Arm Developer without a license. They are available for Linux hosts only. The following FVPs are available:
    - Foundation Platform. This is a basic FVP with a minimal peripheral set. It includes a single cluster which can be configured with 1-4 AEMvA cores. It is suitable for running bare-metal applications and for booting Linux. It supports the CADI debug interface, but does not support MTI or Iris interfaces. It is documented in the Foundation Platform User Guide.
    - Armv-A Base Platform RevC FVP. This is a platform model with a more extended peripheral set than the Foundation Platform. It has two clusters, each of which can be configured with

1-4 AEMvA cores. It supports Arm®v8-A architecture versions up to v8.7 and Armv9-A. It also supports CADI, MTI, and Iris debug and trace interfaces.

- Arm®v8-R AEM FVP. This FVP includes a single cluster of 1-4 AEMv8-R cores. It allows you to target AArch32 or AArch64, RAS, VFP, EL2, and other Arm®v8-R features.
- Armv-A Compliance FVP. This FVP is optimized for validating CPU implementations and can be used with the A-profile Architecture Compliance Kit (ACK) to demonstrate compliance with the Arm® architecture specification.

- Free-of-charge Arm Ecosystem FVPs, including Reference Design FVPs and the Morello Platform FVP can be downloaded from Arm Ecosystem FVPs on Arm Developer.

Related information

Contents of the FVP Standard Library package on page 16
Base Platform
Microcontroller Prototyping System 2
Versatile Express Model

3 Getting Started with Fixed Virtual Platforms

This chapter describes how to use FVPs.

3.1 Contents of the FVP Standard Library package

The FVP Standard Library consolidates commonly used FVPs into a single package which also contains some useful plug-ins and utilities.


Note

The package does not include unlicensed FVPs. These are available for download separately, from Arm Architecture Models on Arm Developer.

The package installs the FVPs under the `models` directory. It also installs:

Plug-ins

Plug-ins are DLLs or shared objects that provide extra functionality to FVPs, for instance different types of trace output. To load a plug-in, pass the name of the plug-in to the FVP at startup using the `--plugin` command-line option or using the `FM_TRACE_PLUGINS` environment variable. For more information about plug-ins, see Plug-ins for Fast Models in the Fast Models Reference Manual.

Model Shell and Model Debugger

Model Shell is a command-line tool for launching FVPs. For more information about Model Shell, see Model Shell for Fast Models Reference Manual. Model Debugger is an easy to use symbolic debugger with a GUI that allows you to debug software running on the FVP. For more information about Model Debugger, see Model Debugger for Fast Models User Guide. They are installed in the `bin` directory.

iris.debug Python module

iris.debug is a Python scripting interface to Fast Models. It allows you to interact with FVPs, including connecting to and configuring them, performing execution control, and accessing registers and memory. It is installed under the `Iris` directory. For more information about iris.debug, see Iris Python Debug Scripting User Guide.

3.2 FVP command-line options

Specify these options when you launch an FVP from the command line. You can specify these options in any order.

Table 3-1: CADI or Iris-related options

Short form	Long form	Description
-S	*--cadi-server*	Start a CADI server. This option allows a CADI-enabled debugger to connect to targets in the simulation. To shut down the server, return to the command window that you used to start the model and press **Ctrl+C**.
-I	*--iris-server*	Start an Iris server. This option allows an Iris-enabled debugger to connect to targets in the simulation.
-R	*--run*	Run the simulation immediately after the CADI or Iris server is started. Use this option with *--cadi-server* or *--iris-server*. The default is to wait until the debugger has connected before running.
-L	*--cadi-log*	Log all CADI function calls made during the simulation into XML files. One log file is created for each CADI target. The log files are created in the current working directory. The filename format is: `CADIlog-<`*`TargetInstanceName`*`>-<`*`ProcessID`*`>.xml`
-i	*--iris-log*	Log to stdout all Iris function calls that were made during the simulation. There are 5 possible log levels. To set a level greater than 1, specify the option multiple times, for example `-ii` for level 2. The log levels have the following meanings: **0** Logging is disabled. This value is the default. **1** Log messages use a compact single-line format. **2** Log messages use a single-line pseudo-JSON format. **3** Log messages use a more readable, multi-line pseudo-JSON format. **4** As 3 but also prints the U64JSON hex value of the message. **Note:** To set the Iris log level for all FVP invocations, use the `IRIS_GLOBAL_INSTANCE_LOG_MESSAGES` environment variable.
-A	*--iris-allow-remote*	Start an Iris server and allow connections to it from a remote workstation. The default is disallowed.

Short form	Long form	Description
-p	*--print-port-number*	Print the port number on which the Iris or CADI server is listening. Use this option with *--cadi-server* or *--iris-server*. **Tip:** This option can be useful if you need to specify the port number when you connect a client to the debug server.
	--iris-port n	Set a port to use for the Iris server. Use this option with *--iris-server*. The default is 7100.
	--iris-port-range min:max	Set the range of ports to scan when starting an Iris server. The server uses the first available port in the range. Use this option with *--iris-server*.

Table 3-2: Output-related options

Short form	Long form	Description
	--list-instances	Print a list of model instances to standard output, then exit the simulation. Use this option to help identify the correct syntax for configuration files, and to find out what instances the target supplies.
-l	*--list-params*	Print a list of model parameters to standard output, then exit the simulation. **Tip:** If you are loading a plug-in, this option also lists the plug-in parameters.
	--dump-params	Dump the list of model parameters into a JSON file called `parameter_list.json`, then exit the simulation. The file is created in the current working directory.
	--list-regs	Print model register information to standard output, then exit the simulation.
	--check-regs	Same as *--list-regs* but with extra consistency checks on the CADI register API.
-o	*--output file*	Redirect output from the `--list-instances`, `--list-memory`, `--list-params`, and `--list-regs` commands to a file. If this option is used with `--list-params`, the contents of the output file are formatted correctly for use as input by the `--config-file` option.
	--log filename	Log all SystemC reports into *filename*.

Short form	Long form	Description
	--stat	Print the following performance statistics on simulation exit: **Simulated time** An estimate of the time that the workload would have taken on the modeled hardware. **User time** Time in wall clock seconds that the host CPU spent running in user mode. **System time** Time in wall clock seconds that the host CPU spent running in system mode. **Wall time** Time in wall clock seconds between the simulation starting and stopping. **Performance index** An estimate of the accuracy of the simulation performance. This value is Simulated time divided by Wall time.
-P	*--prefix*	Prefix each line of semihosting output with the name of the target instance.
-h	*--help*	Print the help message and exit.
	--version	Print version information for the FVP.
-q	*--quiet*	Suppress informational output.
-K	*--keep-console*	Keep the console window open after completion. This option applies to Windows only.
	--disable-mod\el-exitcode	Disable the simulation from retrieving the exit code returned by a model or a plug-in. By default, it is enabled.

Table 3-3: Run control options

Short form	Long form	Description
	--cpulimit n	Maximum number of wall-clock seconds for the simulation process to be active. This value excludes simulation startup and shutdown. This option is ignored if a debug server is started. The default is unlimited.
	--cyclelimit n	Maximum number of cycles to run. This option is ignored if a debug server is started. The default is unlimited.
-T	*--timelimit n*	Maximum number of wall-clock seconds for the simulation to run, excluding startup and shutdown. To terminate the model immediately after initialization, specify `--timelimit 0`.
	--simlimit n	Maximum number of seconds to simulate. This option is ignored if a debug server is started. The default is unlimited. Like the `Simulated time` value output by *--stat*, this value is measured in simulation seconds, not wall-clock seconds.

Short form	Long form	Description
-b	*--break* [*in\stance=*] *ad\dress*	Set a program breakpoint on the address of an instruction. This option can be specified multiple times. If the FVP has multiple cores you must specify an instance, for example: `-b FVP_Base_AEMvA.cluster0.cpu0=0x800010eC`

Table 3-4: Timing and performance options

Short form	Long form	Description
	--cpi-file filename	Use *filename* to set the Cycles Per Instruction (CPI) class. For information about CPI files, see Timing Annotation.
-Q	*--quantum n*	Number of ticks to simulate for each quantum. The default is 10000.
-M	*--min-sync-la\tency n*	Number of ticks to simulate before synchronizing. Events that occur at a higher frequency than this value are missed. The default is 100.
	--fast-ram filename	Enable FastRAM and load the configuration from *filename*. For more information about FastRAM, see FastRAM.

Table 3-5: Memory-related options

Short form	Long form	Description
	--dump file@ad\dress,size	Dump a section of memory to a file at model shutdown. This option can be specified multiple times. The full syntax is: `.--dump [instance=]file@[memspace:]address,size` **Tip:** To see the list of instances and memory spaces, use the *--list-memory* option.
	--data file@address	Write raw data contained in *file* to the specified address. This option can be specified multiple times. The full syntax is: `--data [instance=]file@[memspace:]address`
	--list-memory	Print model memory information to standard output, then exit the simulation.
	--start [*instance=*] *address*	Set the initial PC value to this address, overriding the `.axf` start address. **Note:** • Use this option if you do not want the CPU to start executing at the default reset address. You do not normally need to do this if you are loading an ELF file using *--application*. • This option can be used with *--data* to load binary data that is not in an ELF file.

Table 3-6: Configuration options

Short form	Long form	Description
-C	*--parameter instance.parameter=value*	Set a parameter. This option can be specified multiple times. Specify the full hierarchical name of the parameter. This option is also used to set plug-in parameters.
-f	*--config-file filename*	Load the parameters from a configuration file.

Table 3-7: Options for loading a plug-in or application

Short form	Long form	Description
-a	*--application [instance=] file\name*	Load an application. On a multi-core system, specify the instance, or use `*` to load the application image into all the cores in a cluster: `-a cluster0.cpu*=file`
	--plugin file\name	Load the plug-in *filename*. This option can be specified multiple times. You can also load plug-ins using the `FM_TRACE_PLUGINS` environment variable. For information about plug-ins, see Plug-ins for Fast Models.
	--trace-plugin filename	Load a trace plug-in. **Note:** This option is deprecated. Use *--plugin* instead.

3.3 Loading and running an application on an FVP

There are different ways to launch an FVP, for example from the command prompt, or from Model Debugger or Arm® Development Studio.

To run an FVP from the command prompt, enter the model name followed by the model options. To see all available options, use the *--help* option. This is a list of some of the commonly used options for FVPs:

-a [*instance*=]*filename*.axf

Specifies an application to load, and optionally, the instance or instances to load it on. The file can be in one of the following formats, or in a gzip-compressed version:

- ELF.
- Motorola S-Record.

If the FVP contains multiple core instances, you can specify the instance to load the image on. The instance name can include a wildcard (*) to load the same application image into multiple cores, for example:

```
./FVP_Base_AEMvA -a cluster0.cpu*=__image.axf
```

Omitting the instance name loads the application on all cores in the first cluster. If the FVP has multiple cores but no clusters, you must specify the instance name.

`--data filename.bin@address`

Loads binary data into memory at the address specified.

-C *instance.parameter=value*

Sets a single model parameter. Parameters are specified using a path that separates the instance names and the parameter using dots. For example, `-c bp.flashloader0.fname=fip.bin`. Here, `bp` and `flashloader0` are instance names and `fname` is the parameter. To set multiple parameters using a configuration file, use the *`-f`* option instead. To list all the available parameters, with their type, default value, and description, run the model with the *`--list-params`*, or *`-l`* option.

-f *config_file*.txt

Specifies the name of a plain text configuration file. Configuration files simplify managing multiple model parameters. You can set the same parameters using this option as with the `-c` option.

-S

Starts a CADI debug server. This option allows a CADI-enabled debugger, such as Model Debugger or Arm® Development Studio Debugger, to connect to the running model. By default, the model waits for the debugger to connect before starting.

Related information

Arm Development Studio Getting Started Guide
Using Model Debugger

3.4 Configuring the model

When you start the model from the command line, you can configure it using either:

- One or more *`-c`* command-line arguments.
- A configuration file and the *`-f`* command-line argument.

Each *`-c`* command-line argument or line in the configuration file must contain:

- The name of the component instance.
- The parameter to modify.
- Its value.

Use the following format:

```
instance.parameter=value
```

The `instance` can be a hierarchical path, with each level separated by a dot `.` character.


Note

- Comment lines in the configuration file begin with a `#` character.
- You can set Boolean values using either `true` or `false`, or `1` or `0`.

You can generate a configuration file with all parameters set to default values by using the `-o` option to redirect the output from the `--list-params` option, for example:

```
FVP_Base_AEMvA.exe --list-params -o params.txt
```

3.5 FVP debug

This section describes how to debug an FVP.

FVP debug options

To debug an FVP, you can either:

- Run the FVP from within a CADI-enabled debugger.
- Start the FVP with the *-s* command-line argument and then connect a CADI-enabled debugger to it.

For information about using your debugger in these ways, see your debugger documentation.

Semihosting support

Semihosting enables code running on a platform model to directly access the I/O facilities on a host computer. Examples of these facilities include console I/O and file I/O.

The simulator handles semihosting by intercepting `HLT 0xF000`, `SVC 0x123456`, or `SVC 0xAB`, depending on whether the processor is in A64, A32 or T32 state. It handles all other `HLT`s and `SVC`s as normal.

If the operating system does not use `HLT 0xF000`, `SVC 0x123456`, or `SVC 0xAB` for its own purposes, it is not necessary to disable semihosting support to boot an operating system.

To temporarily or permanently disable semihosting support for a current debug connection, see your debugger documentation.

Related information

Semihosting for AArch32 and AArch64
Using semihosting to access resources on the host computer

3.6 Using the CLCD window

When an FVP starts, the CLCD window opens, representing the contents of the simulated color LCD frame buffer. It automatically resizes to match the horizontal and vertical resolution that is set in the CLCD peripheral registers.

MPS2 FVPs

The LEDs and MCC switches in the CLCD window for MPS2-based FVPs correspond to the LEDs and switches on the physical board. They are controlled by the software that you load onto the FVP. For information on how to use them, see User switches and user LEDs in the Arm® MPS2 and MPS2+ FPGA Prototyping Boards TRM on Arm Developer.

Base Platform and VE FVPs

The top section of the CLCD window for Base Platform and VE FVPs displays the status information:

Figure 3-1: Base Platform CLCD window in its default state at startup

USERSW

Eight white boxes show the state of the User DIP switches.

These represent switch S6 on the VE hardware, USERSW[8:1], which is mapped to bits [7:0] of the SYS_SW register at address `0x1C010004`.

The switches are in the off position by default. To change its state, click in the area above or below a white box.

BOOTSW

Eight white boxes show the state of the VE Boot DIP switches.

These represent switch S8 on the VE hardware, BOOTSEL[8:1], which is mapped to bits [15:8] of the SYS_SW register at address `0x1C010004`.

The switches are in the off position by default.

Note

Changing Boot DIP switch positions while the model is running can result in unpredictable behavior.

S6LED

Eight colored boxes indicate the state of the VE User LEDs.

These represent the red/yellow/green LEDs on the VE hardware, which are mapped to bits [7:0] of the SYS_LED register at address `0x1C010008`.

Daughter

Eight white boxes show the state of the daughterboard DIP switches and eight colored boxes show the state of the daughterboard LEDs.

Total Instr

A counter showing the total number of instructions executed.

Because the FVP models provide a *Programmer's View* (PV) of the system, the CLCD displays total instructions rather than total processor cycles. Timing might differ substantially from the hardware because:

- Bus fabric is simplified.
- Memory latencies are minimized.
- Cycle approximate processor and peripheral models are used.

In general, bus transaction timing is consistent with the hardware, but the timing of operations within the model is not accurate.

Total Time

A counter showing the total elapsed time, in seconds.

This time is wall clock time, not simulated time.

Rate Limit

A feature that disables or enables fast simulation.

Because the system model is highly optimized, your code might run faster than it would on real hardware. This effect might cause timing issues.

Rate Limit is enabled by default. Simulation time is restricted so that it more closely matches real time.

To disable or enable Rate Limit, click the square button. You can configure this option when instantiating the model with the `rate_limit-enable` visualization component parameter.

When you click the **Total Instr** item in the CLCD, the display toggles to show the following:

Instr/sec

The number of instructions that execute per second of wall clock time.

Perf Index

The ratio of real time to simulation time. The larger the ratio, the faster the simulation runs. If you enable the Rate Limit feature, the Perf Index approaches unity.

You can reset the simulation counters by resetting the model.

The FVP CLCD displays the core run state for each core on each cluster using a colored icon. The icons are to the left of the **Total Instr** (or **Inst/sec**) item.

Figure 3-2: Core run state icons for a dual-cluster, quad-core model

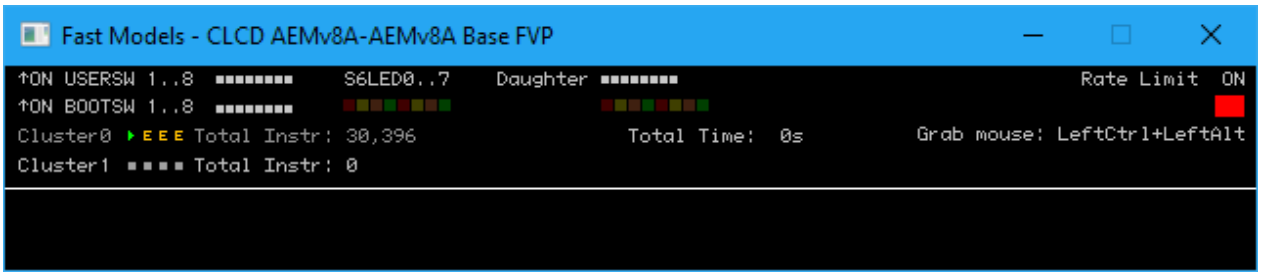


Table 3-8: Core run state icon descriptions

Icon	State label	Description
	UNKNOWN	Run status unknown, that is, simulation has not started.
	RUNNING	Core running, is not idle, and is executing instructions.
	HALTED	External halt signal asserted.
	STANDBY_WFE	Last instruction executed was WFE and standby mode has been entered.
	STANDBY_WFI	Last instruction executed was WFI and standby mode has been entered.
	IN_RESET	External reset signal asserted.
	DORMANT	Partial core power down.
	SHUTDOWN	Complete core power down.

If the CLCD window has focus:

- Any keyboard input is translated to PS/2 keyboard data.
- Any mouse activity over the window is translated into PS/2 relative mouse motion data. The data is then streamed to the KMI peripheral model FIFOs.


Note

The simulator only sends relative mouse motion events to the model. As a result, the host mouse pointer does not necessarily align with the target OS mouse pointer.

You can hide the host mouse pointer by pressing the **left Ctrl**+**left Alt** keys. Press the keys again to redisplay the host mouse pointer. Only the **left Ctrl** key is operational. The **right Ctrl** key does not have the same effect.

If you prefer to use a different key, configure it with the `trap_key` visualization component parameter.

Related information

VEVisualisation component

3.7 Ethernet with VE FVPs

This section describes how to use Ethernet with VE FVPs.

Using Ethernet with VE FVPs

The VE FVPs have a virtual Ethernet component. This component is a model of the SMSC 91C111 Ethernet controller, and uses a TAP device to communicate with the network. By default, the Ethernet component is disabled.

Host requirements

Before you can use the Ethernet capability of VE FVPs, set up your host computer.

Target requirements

This section describes the target requirements.

Target requirements - about

The VE FVPs include a software implementation of the SMSC 91C111 Ethernet controller. Your target OS must therefore include a driver for this specific device. To use the SMSC chip, configure the kernel. Linux supports the SMSC 91C111.

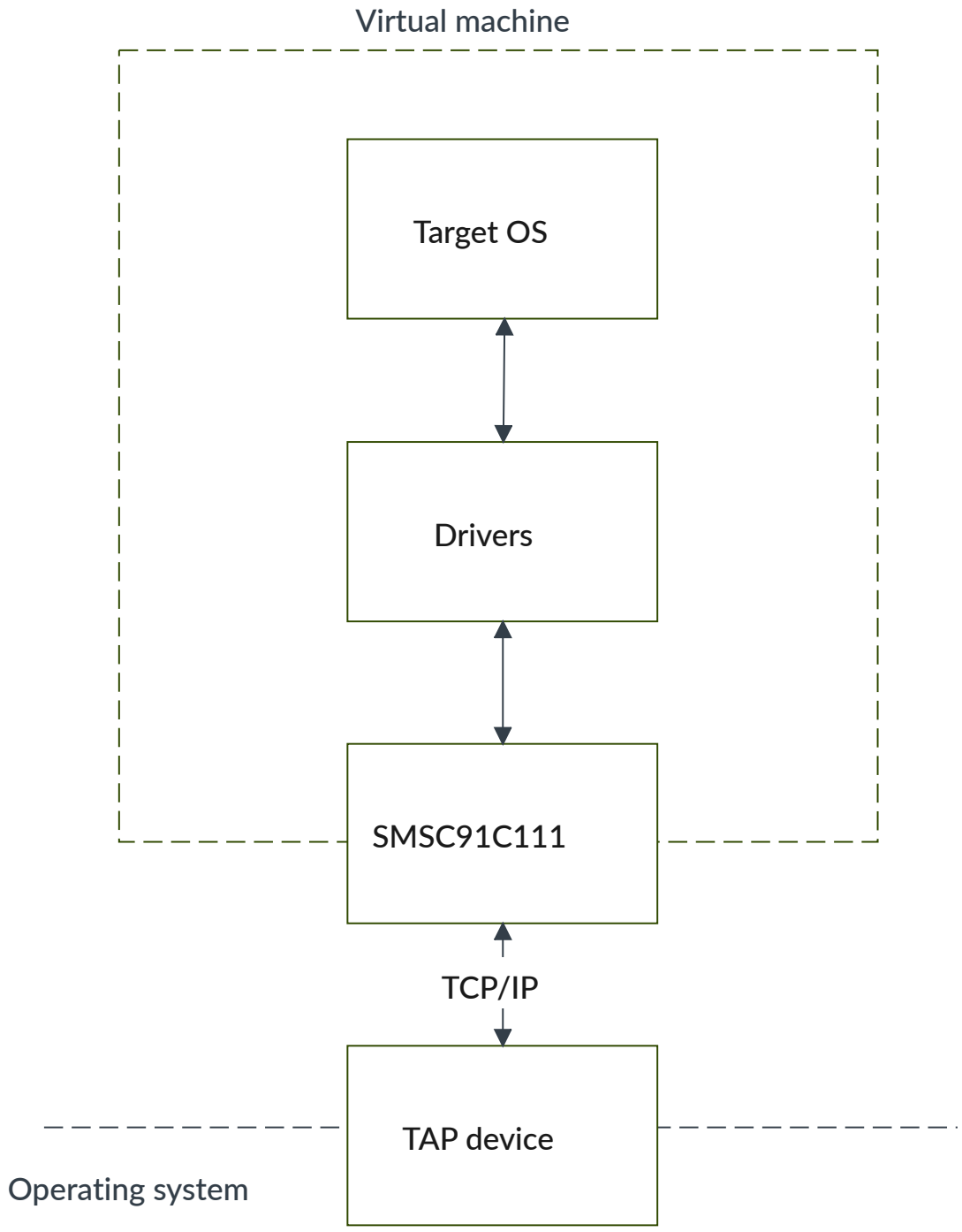
The configurable SMSC 91C111 component parameters are:

- `enabled`.
- `mac_address`.
- `promiscuous`.

enabled

When the device is disabled, the kernel cannot detect the device.

Figure 3-3: Model networking structure block diagram



To perform read and write operations on the TAP device, configure a HostBridge component. The HostBridge component is a virtual *Programmer's View* (PV) model. It acts as a networking

gateway to exchange Ethernet packets with the TAP device on the host, and to forward packets to NIC models.

mac_address

There are two options for the `mac_address` parameter.

If a MAC address is not specified, when the simulator is run it takes the default MAC address, which is randomly generated. This random generation provides some degree of MAC address uniqueness when running models on multiple hosts on a local network.

promiscuous

The Ethernet component starts in promiscuous mode by default. In this mode, it receives all network traffic, even any not addressed to the device. Use this mode if you are using a single network device for multiple MAC addresses. Use this mode if, for example, you share the network card between your host OS and the VE FVP Ethernet component.

By default, the Ethernet device on the VE FVP has a randomly generated MAC address and starts in promiscuous mode.

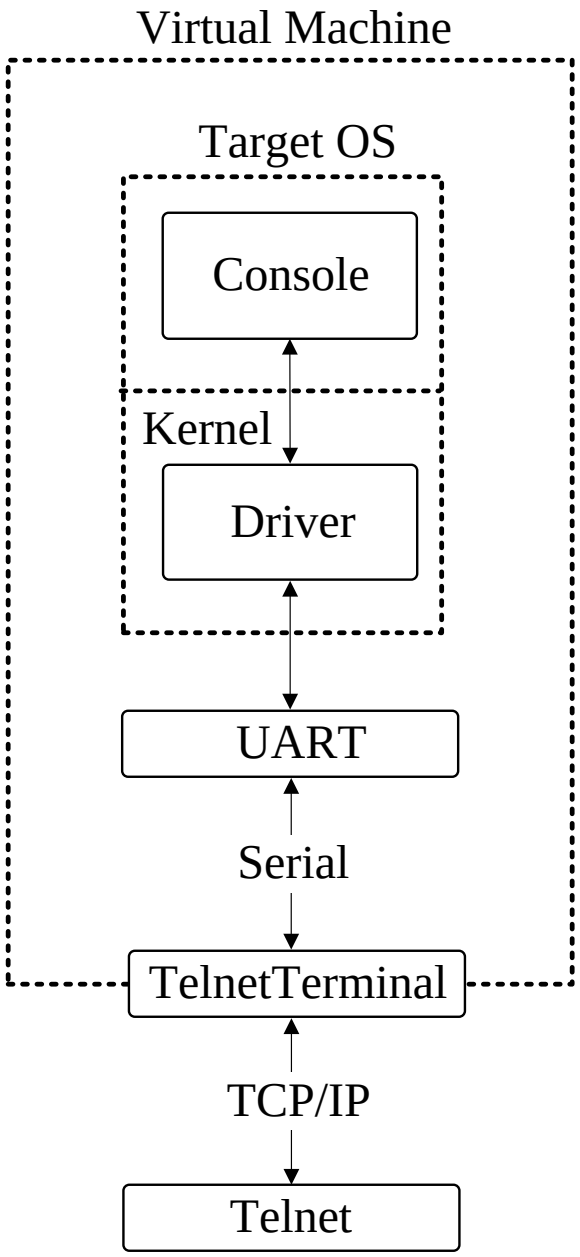
3.8 Using a terminal with a system model

The TelnetTerminal component is a virtual component that enables UART data to be transferred between a TCP/IP socket on the host and a serial port on the target.

Using TelnetTerminal

The following figure shows a block diagram of one possible relationship between the target and host through the TelnetTerminal component. The TelnetTerminal block is what you configure when you define Terminal component parameters. The Virtual Machine is an FVP.

Figure 3-4: Terminal block diagram



On the target side, the console process that is invoked by your target OS relies on a suitable driver being present. Such drivers are normally part of the OS kernel. The driver passes serial data through a UART. The data is forwarded to the TelnetTerminal component. When the simulation is started and the TelnetTerminal component is enabled, the component opens a server (listening) socket on a TCP/IP port. This is port 5000 by default. This port can be connected to by, for example, a Telnet process on the host.

When data becomes available on the network socket, the TelnetTerminal component buffers the data, which can then be read from SerialData.

If there is no connection to the network socket when the first data access is made, and the `start_telnet` parameter is true, a host Telnet session is started automatically. Prior to this first access, you can connect a client of your choice to the network socket.

If the connection between the TelnetTerminal component and the client is broken at any time, for example by closing a client Telnet session, the port is re-opened on the host, permitting you to make another client connection. This could have a different port number if the original one is no longer available.

The port number of a particular TelnetTerminal instance can be defined when your model system starts. The actual value of the port used by each TelnetTerminal is declared when it starts or restarts, and might not be the value that you specified if the port is already in use. If you are using Model Shell, the port numbers are displayed in the host window in which you started the model.


Note

Microsoft Windows 10 disables the Telnet client by default. Follow these steps to enable it:

1. Select **Start** > **Settings**.
2. In the search box, type **Turn Windows features on or off**. The Windows Features dialog opens.
3. Select the **Telnet Client** check box and click **OK**. The installation might take several minutes to complete.

TelnetTerminal parameters

To set the parameters, the syntax to use in a configuration file or on the command line is:

`motherboard.terminal_`*`x`*`.`*`parameter`*`=`*`value`*

where *x* is the terminal identifier and can be 0, 1, 2, or 3.

You can start the TelnetTerminal component in either of the following modes, depending on the `mode` parameter:

`telnet`
: In Telnet mode, the terminal component supports a subset of the RFC 854 protocol. This means that the terminal participates in negotiations between the host and client concerning what is and is not supported, but there is no flow control.

`raw`
: In raw mode the byte stream passes unmodified between the host and the target. The terminal does not participate in initial capability negotiations between the host and client. Instead it acts as a TCP/IP port. You can use this feature to directly connect to your target through the TelnetTerminal component. This permits a debugger connection, for example, to connect a gdb client to a gdbserver running on the target operating system.

The `terminal_command` parameter specifies the command line used to launch a terminal application and connect to the opened TCP port. The TelnetTerminal component replaces the keywords `%port` and `%title`, if specified, with the opened port number and component name, respectively. After replacing `%port` and `%title`, the command line is executed verbatim.

An empty string, which is the default, launches xterm on Linux or `telnet.exe` on Windows.

If you specify a non-empty string, it must include `%port`, but `%title` is optional.

For example:

```
motherboard.terminal_0.terminal_command="putty.exe -telnet %port localhost"
```

3.9 Virtio P9 device component

The VirtioP9Device component is included in Base, BaseR, and A-profile VE platforms. It implements a subset of the Plan 9 file protocol over a virtio transport. It enables accessing a directory on the host's filesystem within Linux, or another operating system that implements the protocol, running on a platform model.

Take the following steps to set up this component:

- Use a version of Linux that supports v9fs over virtio and virtio-mmio devices.
- Update the device tree to include the VirtioP9Device component, or specify it on the kernel command-line, as shown below. The address range for both VE and Base platforms is `0x1C140000-0x1C14FFFF`.
  The interrupt number is 43, or IRQ 75, for both VE and Base platforms.
- Set the following parameter to the directory on the host that you want to mount in the model:

  VE:

  ```
  motherboard.virtiop9device.root_path
  ```

  Base:

  ```
  bp.virtiop9device.root_path
  ```
- On Linux, mount the host directory by using the following command in the model:

  ```
  $ mount -t 9p -o trans=virtio,version=9p2000.L FM <mount point>
  ```

Example kernel command-line argument:

```
virtio_mmio.device=0x10000@0x1c140000:75
```

Example entry for DTS files, to add next to the corresponding `virtio_block` entry:

```
virtio_p9@0140000 {
       compatible = "virtio,mmio";
       reg = <0x0 0x1c140000 0x0 0x1000>;
       interrupts = <0x0 0x2b 0x4>;
};
```

4 Arm® Neoverse™ N1 edge and Neoverse™ E1 edge reference design FVPs

This chapter describes the Arm® Neoverse™ N1 edge and Arm® Neoverse™ E1 edge reference design FVPs. These FVPs are collectively referred to as RD-N1-E1-edge FVPs.

A reference design is a collection of resources, including documentation, a software stack, and FVPs, that describe and model the design choices and performance for recommended configurations of a typical Arm®-based subsystem.

The RD-N1-E1-edge FVPs drive system architecture and software standardization. They provide software and binaries of proprietary firmware that reduce the amount of work that is required for SoC development.

Arm is working to make more content available for these FVPs. To find out more about reference designs, or to contact Arm about them, see Reference Design.

4.1 About the RD-N1-E1 FVPs

The RD-N1-E1-edge FVPs model many of the Arm® IP components in the RD-N1-E1-edge design.

The RD-N1-E1-edge FVPs model the following RD-N1-E1-edge configurations:

Config1
: N1 2xMP4, 512KB L2 cache per core, 4x2 mesh, 2xDMC.

Config2
: E1 2xMP8, 256KB L2 cache per core, 4x2 mesh, 2xDMC.

Config3
: Dual-chip. Two Config1 subsystems linked by CMN-600 CML.

The diagrams in 4.2 Block diagrams for RD-N1-E1-edge on page 35 show the IP components that RD-N1-E1-edge describes. Not all of these components are modeled by these FVPs. The following components are modeled:

- N1 2xMP4 or E1 2xMP8 cores.
- System Control Processor (SCP).
- Management Control Processor (MCP).
- CMN-600.
- Multiple NIC-450 interconnects, although NIC-450 is replaced with a simple bus model.
- Memory access path towards DRAM, including DMC-620 memory controllers.

The following components are not modeled:

- CoreSight™ SoC.
- ELA-500.
- USB.

4.2 Block diagrams for RD-N1-E1-edge

The following diagrams show the composition of RD-N1-E1-edge systems.

Figure 4-1: Block diagram for Neoverse N1 edge reference design

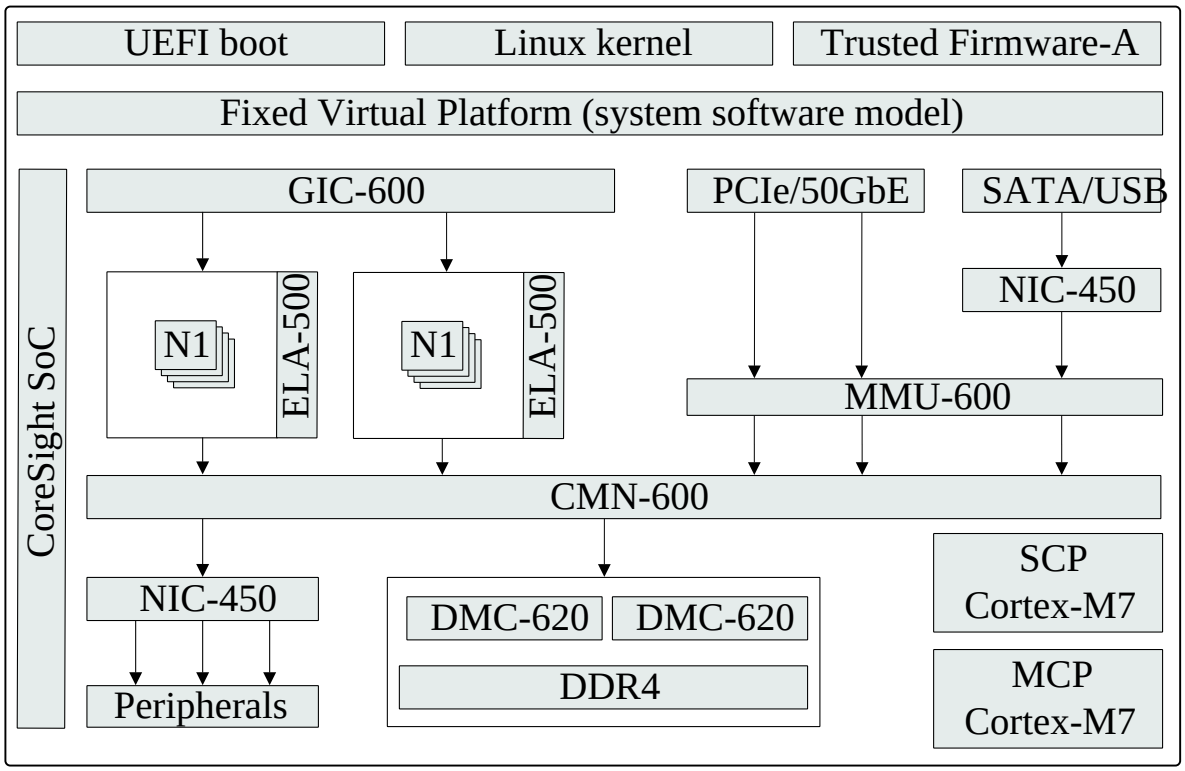
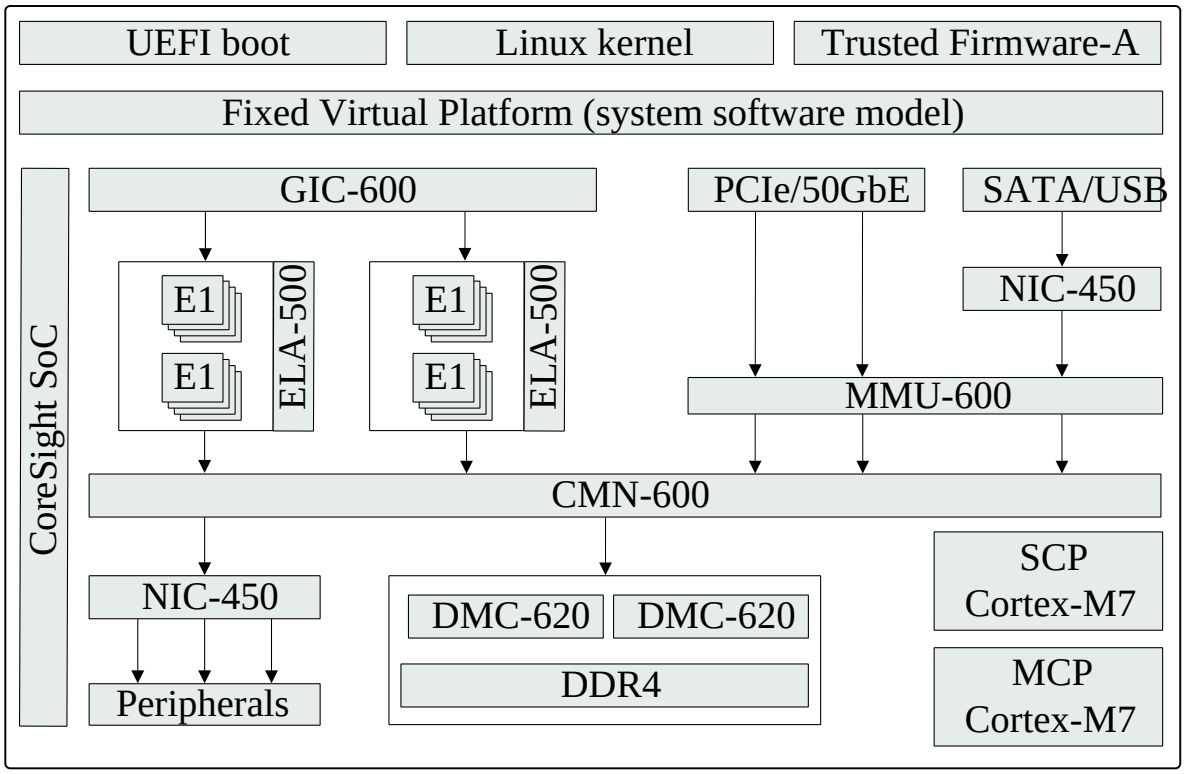


Figure 4-2: Block diagram for Neoverse E1 edge reference design



4.3 RD-N1-E1 FVP peripherals

The RD-N1-E1-edge FVPs include peripherals that the software payload requires to run.

These peripherals are organized in two layers:

SoC

The SoC peripherals represent peripherals that may be added to a compute subsystem in a SoC design.

Board

The board peripherals represent peripherals that may be present on the board onto which the SoC is mounted.

The RD-N1-E1-edge SoC model and board model are based on the Juno Arm® Development Platform (ADP).

In Config3, each compute subsystem has its own SoC and board layers. In this two-chip configuration, each chip is assigned the following memory regions:

Chip 0

`0x000_0000_0000-0x3FF_FFFF_FFFF`

Chip 1

`0x400_0000_0000-0x7FF_FFFF_FFFF`

A sideband communication channel is required to coordinate multi-chiplet software boot over CMN-600. The FVP implements this using the MHU device, but Arm recommends using a solution such as I2C in hardware.

4.3.1 Memory map for RD-N1-E1-edge FVP SoC peripherals

This table shows the memory map for the SoC peripherals in the RD-N1-E1-edge FVPs.

The SoC peripherals area in the RD_N1_E1_edge memory map is mapped to an Expansion AXI region. Therefore, these mappings are not defined in the reference design.

Table 4-1: SoC peripherals memory map

Name	Base address	Size	Description
SMC interface	`0x00_0800_0000`	368MB	Routed to board
SMMUv3	`0x00_2B40_0000`	1MB	-
PCIe config	`0x00_6000_0000`	16MB	-
PCIe memory	`0x00_7000_0000`	132MB	-
DMA MMU-400	`0x00_7FB0_0000`	64KB	-
HDLCD1 MMU-400	`0x00_7FB1_0000`	64KB	-
HDLCD0 MMU-400	`0x00_7FB2_0000`	64KB	-
DDR3 PHY 0	`0x00_7FB6_0000`	64KB	Dummy APB
DDR3 PHY 1	`0x00_7FB7_0000`	64KB	Dummy APB
DDR3 PHY 2	`0x00_7FB8_0000`	64KB	Dummy APB
DDR3 PHY 3	`0x00_7FB9_0000`	64KB	Dummy APB
SoC interconnect NIC-400 GPV	`0x00_7FD0_0000`	1MB	-
Surge detector	`0x00_7FE5_0000`	4KB	Dummy APB
TRNG	`0x00_7FE6_0000`	4KB	-
Trusted non-volatile counters	`0x00_7FE7_0000`	4KB	-
Trusted Root-Key storage	`0x00_7FE8_0000`	4KB	-
Secure I2C	`0x00_7FE9_0000`	256B	A Secure I2C component does not exist in the FVP. Instead, a PL061_GPIO is mapped as a dummy component. It is not functional as a GPIO.
DMA non-secure	`0x00_7FF0_0000`	4KB	-

Name	Base address	Size	Description
DMA secure	0x00_7FF1_0000	4KB	-
HDLCD1	0x00_7FF5_0000	4KB	-
HDLCD0	0x00_7FF6_0000	4KB	-
UART 1	0x00_7FF7_0000	4KB	-
UART 0	0x00_7FF8_0000	4KB	-
I2S	0x00_7FF9_0000	1KB	An I2S component does not exist in the FVP. Instead, a PL061_GPIO is mapped as a dummy component. It is not functional as a GPIO.
I2C	0x00_7FFA_0000	0x256B	An I2C component does not exist in the FVP. Instead, a PL061_GPIO is mapped as a dummy component. It is not functional as a GPIO.
PL352	0x00_7FFD_0000	4KB	PL354
AP configuration	0x00_7FFE_0000	4KB	GPR
System override registers	0x00_7FFF_0000	4KB	-
PCIe memory	0x05_0000_0000	12GB	-

Note

The following devices are discoverable on the PCIe bus:

- Virtio block device x 2.
- AHCI controller with attached SATA disk.

4.3.2 Memory map for RD-N1-E1-edge FVP board peripherals

This table shows the memory map for the board peripherals in the RD-N1-E1-edge FVPs.

Note

The board peripherals area in the RD_N1_E1_edge memory map is mapped to an Expansion AXI region. Therefore, these mappings are not defined in the reference design.

Table 4-2: Board peripherals memory map

Name	Base address	Size	Description
NOR Flash 0	0x00_0800_0000	64MB	-
NOR Flash 1	0x00_0C00_0000	64MB	-
NOR Flash 2	0x00_1000_0000	64MB	-
Ethernet	0x00_1800_0000	64MB	SMSC 91C111
System registers	0x00_1C01_0000	64KB	-
MCI	0x00_1C05_0000	64KB	PL180
KMI 0	0x00_1C06_0000	64KB	PL050
KMI 1	0x00_1C07_0000	64KB	PL050
UART 0	0x00_1C09_0000	64KB	PL011
UART 1	0x00_1C0A_0000	64KB	PL011

Name	Base address	Size	Description
Watchdog	0x00_1C0F_0000	64KB	SP805
Dual timer	0x00_1C11_0000	64KB	SP804
Virtio block device	0x00_1C13_0000	64KB	-
Virtio net device	0x00_1C15_0000	64KB	-
RTC	0x00_1C17_0000	64KB	PL031
GPIO 0	0x00_1C1D_0000	64KB	PL061_GPIO is mapped as a dummy component in the FVP. It is not functional as a GPIO.
GPIO 1	0x00_1C1E_0000	64KB	PL061_GPIO is mapped as a dummy component in the FVP. It is not functional as a GPIO.
DRAM	0x00_8000_0000	2GB	-
DRAM	0x80_8000_0000	6GB	-

4.3.3 Interrupt maps for RD-N1-E1-edge FVP

These tables show the interrupt IDs and sources for the FVP peripherals.

Table 4-3: Interrupt map at the SoC layer

Interrupt ID	Source	Description
147	UART 0	-
148	UART 1	-
171	TRNG	-

Table 4-4: Interrupt map at the board layer

Interrupt ID	Source	Description
111	Ethernet	-
132	RTC	-
133	UART 0	-
134	UART 1	-
140	VFS2	-
202	Virtio	-
204	Virtio net device	-
228	Watchdog	-
229	KMI 0	-
230	Dual timer	Interrupts 0 and 1
231	System registers ethernet IRQ	-

5 Arm® Neoverse™ N2 reference design FVP

The Arm® Neoverse™ N2 reference design (RD-N2) Fixed Virtual Platform (FVP) models much of the Arm® IP in RD-N2. The FVP enables partners to develop ahead of hardware availability and to explore the design from a software perspective.

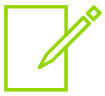
The FVP is used with the RD-N2 software package. For instructions on setting up and running the FVP, see the RD-N2 Software Bundle Readme.

5.1 About the RD-N2 FVP

The RD-N2 FVP models CFG32C4M, which is a single-chiplet system with the number of Arm® Neoverse™ N2 cores reduced to 16.

The FVP models the following IP components:

- Arm® Neoverse™ N2 MP1 cores.
- CMN-700.
- Multiple NIC-450 interconnects.
- GIC-700.
- MMU-700.
- Arm® Cortex®-M7 SCP and MCP cores.


Note

The FVP does not model every component that RD-N2 describes. For example, the FVP does not model the CoreSight™ technology components.

The RD-N2 FVP drives system architecture and software standardization. The models provide software and binaries of proprietary firmware that reduce the amount of work that is required for SoC development.

5.2 RD-N2 FVP peripherals

The RD-N2 FVP includes peripherals that the software payload requires to run.

The SoC peripherals area and board peripherals area in the RD-N2 memory map are mapped to an expansion AMBA® AXI region. Therefore, these mappings are not defined in the reference design.

The peripherals are organized into two layers:

SoC

The SoC peripherals represent peripherals that might be added to a compute subsystem in an SoC design. The RD-N2 SoC model is based on the Juno Arm® Development Platform (ADP).

Board

The board peripherals represent peripherals that might be present on the board onto which the SoC is mounted. The RD-N2 board model is based on the Juno ADP.

5.2.1 Memory map for RD-N2 FVP SoC peripherals

This table shows the memory map for the SoC peripherals in the RD-N2 FVP.

Table 5-1: RD-N2 FVP SoC peripherals

Name	Base address	Size	Description
SMC0 interface	0x00_0800_0000	64MB	Routed to the board.
SMC1 interface	0x10_5000_0000	256MB	Routed to the board.
PCIe config	0x10_1000_0000	256MB	-
PCIe memory	0x00_6000_0000	512MB	-
DMA MMU-400	0x00_E00_0000	64KB	-
HDLCD1 MMU-400	0x00_E01_0000	64KB	-
HDLCD0 MMU-400	0x00_E02_0000	64KB	-
SoC interconnect NIC-400 GPV	0x00_ED0_0000	1MB	-
Surge detector	0x00_EE5_0000	4KB	Dummy APB.
TRNG	0x00_EE6_0000	4KB	-
Trusted non-volatile counters	0x00_EE7_0000	4KB	-
Trusted root-key storage	0x00_EE8_0000	4KB	-
Secure I2C	0x00_EE9_0000	512 bytes	There is no Secure I2C component in the FVP. Instead, a PL061 GPIO is mapped as a dummy component.
DDR PHY 0–31	0x00_EB0_0000	2MB	Dummy APB.
DMA secure	0x00_EF0_0000	64KB	-
DMA non-secure	0x00_EF1_0000	64KB	-
PCIE macro	0x00_EF2_0000	64KB	-
PCIE root port	0x00_EF3_0000	64KB	-
HDLCD1	0x00_EF5_0000	4KB	-
HDLCD0	0x00_EF6_0000	4KB	-
UART 0	0x00_EF7_0000	4KB	-
UART 1	0x00_EF8_0000	4KB	-
I2S	0x00_EF9_0000	1KB	There is no I2S component in the FVP. Instead, a PL061 GPIO is mapped as a dummy component.
I2C	0x00_EFA_0000	256 bytes	There is no I2C component in the FVP. Instead, a PL061 GPIO is mapped as a dummy component.

Name	Base address	Size	Description
PL354	0x00_EFD_0000	64KB	Arm® PrimeCell SMC dual SRAM memory interface (PL354).
System override registers	0x00_EFF_0000	64KB	-
AP configuration	0x00_EFE_0000	64KB	Granular Power Requester (GPR).

5.2.2 Memory map for RD-N2 FVP board peripherals

This table shows the memory map for the board peripherals in the RD-N2 FVP.

Table 5-2: RD-N2 FVP board peripherals

Name	Base address	Size	Description
NOR flash 0	0x00_0800_0000	64MB	-
NOR flash 1	0x10_5000_0000	64MB	-
NOR flash 2	0x10_5400_0000	64MB	-
Ethernet	0x10_5C00_0000	64MB	Non-PCI Ethernet controller (SMSC 91C111)
System registers	0x00_0C01_0000	64KB	-
SP810 Sysctrl	0x00_0C02_0000	64KB	-
MCI	0x00_0C05_0000	64KB	Arm® PrimeCell Multimedia Card Interface (PL180)
KMI 0	0x00_0C06_0000	64KB	Arm® PrimeCell PS2 Keyboard/Mouse Interface (PL050)
KMI 1	0x00_0C07_0000	64KB	Arm® PrimeCell PS2 Keyboard/Mouse Interface (PL050)
UART 0	0x00_0C09_0000	64KB	Arm® PrimeCell UART (PL011)
UART 1	0x00_0C0A_0000	64KB	Arm® PrimeCell UART (PL011)
Watchdog	0x00_0C0F_0000	64KB	Arm® Watchdog Module (SP805)
Dual timer	0x00_0C11_0000	64KB	Arm® Dual-Timer Module (SP804)
Virtio block device	0x00_0C13_0000	64KB	-
Virtio net device	0x00_0C15_0000	64KB	-
GPIO 2Wire (DVI)	0x00_0C16_0000	64KB	Arm® PrimeCell General Purpose Input/Output (PL061)
RTC0	0x00_0C17_0000	64KB	Arm® PrimeCell Real Time Clock (PL031)
RTC1	0x00_0C18_0000	64KB	Arm® PrimeCell Real Time Clock (PL031)
GPIO 0	0x00_0C1D_0000	64KB	Arm® PrimeCell General Purpose Input/Output (PL061)
GPIO 1	0x00_0C1E_0000	64KB	Arm® PrimeCell General Purpose Input/Output (PL061)
DRAM	0x00_8000_0000	2GB	-
DRAM	0x80_8000_0000	6GB	-

5.2.3 Interrupt maps for RD-N2 FVP

These tables show the interrupt IDs and sources for the FVP peripherals.

Table 5-3: Interrupt map at the SoC layer

Interrupt ID	Source
403	UART 0
404	UART 1
427	TRNG

Table 5-4: Interrupt map at the board layer

Interrupt ID	Source
387	RTC1
388	RTC0
389	UART0 (board)
390	UART1 (board)
391	KMI1
392	GPIO0
393	GPIO1
394	I2C GPIO
395	MCIINTR0
396	MCIINTR1
397	SMSC 91C111
405	HDLCD controller 0
406	SMC PL354 interface 0
407	SMC PL354 interface 1
413	HDLCD controller 1
414	SMMU combined Secure interrupt
415	SMMU combined Non-secure interrupt
418–425	DMA0 IRQ7–0
426	DMA0 IRQ abort
428–435	DMA1 IRQ7–0
436	DMA1 IRQ abort
458	Virtio block device
460	Virtio net
483	RTCC
484	WDT
485	KMI0
486	Dual timer
487	System registers
488	System register – USB

Interrupt ID	Source
489	System register – Tile
490	System register – Push button
491	System register – Ethernet

6 Arm® Neoverse™ V1 reference design FVP

To develop ahead of hardware availability and to explore the design from a software perspective, the Fixed Virtual Platform (FVP) models many of the Arm® IP in the RD-V1 design.

6.1 About the RD-V1 FVP

Two configurations of RD-V1 are supported.

- RD-V1 FVP models Config-M, a single-chiplet system with 16 Arm® Neoverse™ V1 cores.
- RD-V1 quad-chiplet FVP models a reduced-size variant of Config-XL, consisting of four compute subsystems linked by CMN-650 CML. It provides a functional model of a quad-chiplet system. Each subsystem contains four Arm® Neoverse™ V1 cores, for a total of 16 cores in the FVP (at full size, Config-XL has 4 x 32 cores).

The FVP models the following IP components:

- Arm® Neoverse™ V1 MP1.
- GIC-700.
- MMU-600.
- SCP.
- MCP.
- CMN-650.
- Multiple NIC-450 interconnects.
- Memory access path towards DRAM which includes a TrustZone® controller.


Note

The FVP does not model every component that RD-V1 describes. For example, it does not model the CoreSight™ technology components.

The RD-V1 FVP drives system architecture and software standardization. The models provide software and binaries of proprietary firmware that reduce the amount of work that is required for SoC development.

6.2 RD-V1 FVP peripherals

The RD-V1 FVP includes peripherals that the software payload requires to run.

These peripherals are organized in two layers:

SoC

The SoC peripherals represent peripherals that can be added to a compute subsystem in a SoC design. The Arm® Neoverse™ V1 reference design SoC model is based on the Juno Arm Development Platform (ADP).

Board

The board peripherals represent peripherals that can be present on the board onto which the SoC is mounted. The RD-V1 board model is based on the Juno Arm Development Platform (ADP).

In the quad-chiplet system, each compute subsystem has SoC and Board layers dedicated to that subsystem. Addressing for each chip is defined in chapter 7.1, AP memory map, in the Arm® Neoverse™ V1 reference design Software Developer Guide.

Chip 0 `0x000_0000_0000-0x3FF_FFFF_FFFF`

Chip 1 `0x400_0000_0000-0x7FF_FFFF_FFFF`

Chip 2 `0x800_0000_0000-0xBFF_FFFF_FFFF`

Chip 3 `0xC00_0000_0000-0xFFF_FFFF_FFFF`

A sideband communication channel is required to coordinate multi-chiplet software boot over CMN-650. The FVP implements this using the MHU device, but Arm recommends using a solution such as I^2C in hardware.

Related information

Arm Neoverse v1 reference design Software Developer Guide

6.2.1 Memory map for RD-V1 FVP SoC peripherals

This table shows the memory map for the SoC peripherals in the RD-V1 FVP.

Note

The following devices are discoverable on the PCIe bus:

- Virtio block device x 2.
- AHCI controller with attached SATA disk.

Table 6-1: SoC peripherals

Name	Base address	Size	Description
SMC interface	`0x00_0800_0000`	368MB	Routed to Board
SMMUv3	`0x00_2B40_0000`	1MB	-
PCIe Config	`0x00_6000_0000`	16MB	-
PCIe Memory	`0x00_7000_0000`	132MB	-
DMA MMU-400	`0x00_7FB0_0000`	64KB	-

Name	Base address	Size	Description
HDLCD1 MMU-400	0x00_7FB1_0000	64KB	-
HDLCD0 MMU-400	0x00_7FB2_0000	64KB	-
DDR3 PHY 0	0x00_7FB6_0000	64KB	Dummy APB
DDR3 PHY 1	0x00_7FB7_0000	64KB	Dummy APB
DDR3 PHY 2	0x00_7FB8_0000	64KB	Dummy APB
DDR3 PHY 3	0x00_7FB9_0000	64KB	Dummy APB
SoC Interconnect NIC-400 GPV	0x00_7FD0_0000	1MB	-
Surge detector	0x00_7FE5_0000	4KB	Dummy APB
TRNG	0x00_7FE6_0000	4KB	-
Trusted Non-volatile counters	0x00_7FE7_0000	4KB	-
Trusted Root-Key storage	0x00_7FE8_0000	4KB	-
Secure I2C	0x00_7FE9_0000	256B	A Secure I2C component does not exist in the FVP. Instead, a PL061_GPIO is mapped.
DMA Non-secure	0x00_7FF0_0000	4KB	-
DMA Secure	0x00_7FF1_0000	4KB	-
HDLCD1	0x00_7FF5_0000	4KB	-
HDLCD0	0x00_7FF6_0000	4KB	-
UART 1	0x00_7FF7_0000	4KB	-
UART 0	0x00_7FF8_0000	4KB	-
I2S	0x00_7FF9_0000	1KB	An I2S component does not exist in the FVP. Instead, a PL061_GPIO is mapped.
I2C	0x00_7FFA_0000	256B	An I2C component does not exist in the FVP. Instead, a PL061_GPIO is mapped.
PL352	0x00_7FFD_0000	4KB	PL354
AP configuration	0x00_7FFE_0000	4KB	GPR
System override Registers	0x00_7FFF_0000	4KB	-
PCIe Memory	0x05_0000_0000	12GB	-

6.2.2 Memory map for RD-V1 FVP board peripherals

This table shows the memory map for the board peripherals in the RD-V1 FVP.


Note

The SoC peripherals area and board peripherals area in the RD-V1 memory map are mapped to an Expansion AXI region. Therefore, these mappings are not defined in the reference design.

Table 6-2: Board peripherals

Name	Base address	Size	Description
NOR Flash 0	0x00_0800_0000	64MB	-
NOR Flash 1	0x00_0C00_0000	64MB	-
NOR Flash 2	0x00_1000_0000	64MB	-
Ethernet	0x00_1800_0000	64MB	SMSC 91C111
System registers	0x00_1C01_0000	64KB	-
MCI	0x00_1C05_0000	64KB	PL180
KMI 0	0x00_1C06_0000	64KB	PL050
KMI 1	0x00_1C07_0000	64KB	PL050
UART 0	0x00_1C09_0000	64KB	PL011
UART 1	0x00_1C0A_0000	64KB	PL011
Watchdog	0x00_1C0F_0000	64KB	SP805
Dual Timer	0x00_1C11_0000	64KB	SP804
Virtio Block Device	0x00_1C13_0000	64KB	-
Virtio Net Device	0x00_1C15_0000	64KB	-
RTC	0x00_1C17_0000	64KB	PL031
GPIO 0	0x00_1C1D_0000	64KB	PL061_GPIO
GPIO 1	0x00_1C1E_0000	64KB	PL061_GPIO
DRAM	0x00_8000_0000	2GB	-
DRAM	0x80_8000_0000	6GB	-

6.2.3 Interrupt maps for RD-V1 FVP

These tables show the interrupt IDs and sources for the FVP peripherals.

Table 6-3: Interrupt map at the SoC layer

Interrupt ID	Source	Description
147	UART 0	-
148	UART 1	-
171	TRNG	-

Table 6-4: Interrupt map at the board layer

Interrupt ID	Source	Description
111	Ethernet	-
132	RTC	-
133	UART 0	-
134	UART 1	-
140	VFS2	-
202	Virtio	-

Interrupt ID	Source	Description
204	Virtio net device	-
228	Watchdog	-
229	KMI 0	-
230	Dual Timer	Interrupts 0 and 1
231	System registers Ethernet IRQ	-

7 Base Platform FVPs

This chapter lists the Base Platform FVPs and the instances in them.

For the Base Platform memory map, see Base Platform memory map in the Fast Models Reference Manual.

7.1 FVP_Base_AEMv8R

FVP_Base_AEMv8R contains the following instances:

FVP_Base_AEMv8R instances

FVP_Base_AEMv8R
: Base Platform Compute Subsystem for AEMv8RMPCT.
Type: `FVP_Base_AEMv8R`.

FVP_Base_AEMv8R.bp
: Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_AEMv8R.bp.Timer_0_1
: ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_AEMv8R.bp.Timer_0_1.clk_div0
: A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.Timer_0_1.clk_div1
: A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.Timer_0_1.counter0
: Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_AEMv8R.bp.Timer_0_1.counter1
: Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_AEMv8R.bp.Timer_2_3
: ARM Dual-Timer Module(SP804).

Type: SP804_Timer.

FVP_Base_AEMv8R.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMv8R.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMv8R.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_AEMv8R.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_AEMv8R.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_AEMv8R.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_AEMv8R.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMv8R.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMv8R.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMv8R.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_AEMv8R.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_AEMv8R.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_AEMv8R.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_AEMv8R.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_AEMv8R.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMv8R.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMv8R.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMv8R.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_AEMv8R.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMv8R.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMv8R.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMv8R.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMv8R.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_AEMv8R.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_AEMv8R.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_AEMv8R.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_AEMv8R.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_AEMv8R.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_AEMv8R.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_AEMv8R.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_AEMv8R.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_AEMv8R.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_AEMv8R.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_AEMv8R.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMv8R.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMv8R.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMv8R.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMv8R.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_AEMv8R.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_AEMv8R.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_AEMv8R.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_AEMv8R.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_AEMv8R.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_AEMv8R.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_AEMv8R.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_AEMv8R.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_AEMv8R.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_AEMv8R.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMv8R.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_AEMv8R.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_AEMv8R.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMv8R.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMv8R.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMv8R.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMv8R.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_AEMv8R.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_AEMv8R.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_AEMv8R.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_AEMv8R.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_AEMv8R.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_AEMv8R.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_AEMv8R.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_AEMv8R.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_AEMv8R.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_AEMv8R.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_AEMv8R.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_AEMv8R.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_AEMv8R.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_AEMv8R.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_AEMv8R.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_AEMv8R.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_AEMv8R.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_AEMv8R.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_AEMv8R.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_AEMv8R.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_AEMv8R.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_AEMv8R.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_AEMv8R.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_AEMv8R.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_AEMv8R.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMv8R.cluster0

ARMAEMv8-R Cluster CT model.
Type: `Cluster_ARMAEMv8-R_MP`.

FVP_Base_AEMv8R.cluster0.cpu0

ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_Base_AEMv8R.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMv8R.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.cpu1

ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_Base_AEMv8R.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMv8R.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.cpu2
ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_Base_AEMv8R.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMv8R.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.cpu3
ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_Base_AEMv8R.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMv8R.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMv8R.cluster0_labeller
Type: `Labeller`.

FVP_Base_AEMv8R.dapmemlogger
Bus Logger.

Type: PVBusLogger.

FVP_Base_AEMv8R.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_AEMv8R.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_AEMv8R.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.2 FVP_Base_AEMvA

FVP_Base_AEMvA contains the following instances:

FVP_Base_AEMvA instances

FVP_Base_AEMvA
Base Platform Compute Subsystem for AEMvACT.
Type: FVP_Base_AEMvA.

FVP_Base_AEMvA.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_AEMvA.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_AEMvA.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_AEMvA.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_AEMvA.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_AEMvA.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_AEMvA.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_AEMvA.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_AEMvA.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_AEMvA.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_AEMvA.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_AEMvA.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_AEMvA.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_AEMvA.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_AEMvA.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMvA.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMvA.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMvA.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMvA.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_AEMvA.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_AEMvA.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_AEMvA.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_AEMvA.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_AEMvA.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_AEMvA.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_AEMvA.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_AEMvA.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_AEMvA.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_AEMvA.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_AEMvA.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMvA.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMvA.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMvA.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_AEMvA.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_AEMvA.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_AEMvA.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_AEMvA.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_AEMvA.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_AEMvA.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_AEMvA.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_AEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_AEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_AEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_AEMvA.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_AEMvA.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_AEMvA.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_AEMvA.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_AEMvA.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_AEMvA.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_AEMvA.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMvA.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMvA.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_AEMvA.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_AEMvA.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_AEMvA.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_AEMvA.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_AEMvA.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_AEMvA.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_AEMvA.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_AEMvA.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_AEMvA.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_AEMvA.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_AEMvA.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_AEMvA.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_AEMvA.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_AEMvA.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_AEMvA.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_AEMvA.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_AEMvA.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_AEMvA.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_AEMvA.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA.cluster0

ARM AEMvA Cluster CT model.
Type: Cluster_ARM_AEM-A_MP.

FVP_Base_AEMvA.cluster0.cpu0

ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_AEMvA.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_AEMvA.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.cpu1
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_AEMvA.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_AEMvA.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.cpu2
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_AEMvA.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_AEMvA.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.cpu3
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_AEMvA.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_AEMvA.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA.cluster0_labeller
Type: Labeller.

FVP_Base_AEMvA.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_AEMvA.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_AEMvA.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_AEMvA.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.3 FVP_Base_AEMvA-AEMvA

FVP_Base_AEMvA-AEMvA contains the following instances:

FVP_Base_AEMvA-AEMvA instances

FVP_Base_AEMvA_AEMvA
Base Platform Compute Subsystem for AEMvACT and AEMvACT.
Type: FVP_Base_AEMvA_AEMvA.

FVP_Base_AEMvA_AEMvA.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_AEMvA_AEMvA.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_AEMvA_AEMvA.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA_AEMvA.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_AEMvA_AEMvA.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_AEMvA_AEMvA.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_AEMvA_AEMvA.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA_AEMvA.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_AEMvA_AEMvA.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_AEMvA_AEMvA.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_AEMvA_AEMvA.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_AEMvA_AEMvA.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_AEMvA_AEMvA.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_AEMvA_AEMvA.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_AEMvA_AEMvA.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_AEMvA_AEMvA.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMvA_AEMvA.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMvA_AEMvA.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMvA_AEMvA.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMvA_AEMvA.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_AEMvA_AEMvA.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_AEMvA_AEMvA.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_AEMvA_AEMvA.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_AEMvA_AEMvA.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_AEMvA_AEMvA.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_AEMvA_AEMvA.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_AEMvA_AEMvA.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_AEMvA_AEMvA.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_AEMvA_AEMvA.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMvA_AEMvA.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_AEMvA_AEMvA.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_AEMvA_AEMvA.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_AEMvA_AEMvA.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_AEMvA_AEMvA.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_AEMvA_AEMvA.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_AEMvA_AEMvA.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_AEMvA_AEMvA.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_AEMvA_AEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_AEMvA_AEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_AEMvA_AEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_AEMvA_AEMvA.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_AEMvA_AEMvA.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_AEMvA_AEMvA.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_AEMvA_AEMvA.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_AEMvA_AEMvA.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_AEMvA_AEMvA.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_AEMvA_AEMvA.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_AEMvA_AEMvA.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_AEMvA_AEMvA.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_AEMvA_AEMvA.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_AEMvA_AEMvA.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_AEMvA_AEMvA.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA_AEMvA.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA_AEMvA.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA_AEMvA.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_AEMvA_AEMvA.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_AEMvA_AEMvA.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_AEMvA_AEMvA.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_AEMvA_AEMvA.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_AEMvA_AEMvA.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_AEMvA_AEMvA.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_AEMvA_AEMvA.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_AEMvA_AEMvA.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_AEMvA_AEMvA.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_AEMvA_AEMvA.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_AEMvA_AEMvA.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_AEMvA_AEMvA.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_AEMvA_AEMvA.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_AEMvA_AEMvA.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_AEMvA_AEMvA.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_AEMvA_AEMvA.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_AEMvA_AEMvA.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_AEMvA_AEMvA.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_AEMvA_AEMvA.cluster0
ARM AEMvA Cluster CT model.
Type: `Cluster_ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu0
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu1
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu2
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu3
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster0_labeller
Type: `Labeller`.

FVP_Base_AEMvA_AEMvA.cluster1
ARM AEMvA Cluster CT model.
Type: `Cluster_ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster1.cpu0
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMvA_AEMvA.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_AEMvA_AEMvA.cluster1.cpu1
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_AEMvA_AEMvA.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_AEMvA_AEMvA.cluster1.cpu1.l1dcache
PV Cache.

Type: PVCache.

FVP_Base_AEMvA_AEMvA.cluster1.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA_AEMvA.cluster1.cpu2
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_AEMvA_AEMvA.cluster1.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_AEMvA_AEMvA.cluster1.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA_AEMvA.cluster1.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA_AEMvA.cluster1.cpu3
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_AEMvA_AEMvA.cluster1.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_AEMvA_AEMvA.cluster1.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA_AEMvA.cluster1.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA_AEMvA.cluster1.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_AEMvA_AEMvA.cluster1_labeller
Type: Labeller.

FVP_Base_AEMvA_AEMvA.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_AEMvA_AEMvA.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_AEMvA_AEMvA.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_AEMvA_AEMvA.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.4 FVP_Base_Cortex-A32x1

FVP_Base_Cortex-A32x1 contains the following instances:

FVP_Base_Cortex-A32x1 instances

FVP_Base_Cortex_A32x1

Base Platform Compute Subsystem for ARMCortexA32x1CT.
Type: FVP_Base_Cortex_A32x1.

FVP_Base_Cortex_A32x1.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A32x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A32x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).

Type: SP804_Timer.

FVP_Base_Cortex_A32x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A32x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A32x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A32x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A32x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A32x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A32x1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A32x1.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A32x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A32x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A32x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A32x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A32x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A32x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A32x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A32x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A32x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A32x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A32x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A32x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A32x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A32x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A32x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A32x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A32x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A32x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A32x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A32x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A32x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A32x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A32x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A32x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A32x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A32x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A32x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A32x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A32x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A32x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A32x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A32x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A32x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A32x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A32x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A32x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A32x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A32x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A32x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A32x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A32x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A32x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A32x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A32x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A32x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A32x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A32x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A32x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A32x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A32x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x1.cluster0

ARM Cortex-A32 Cluster CT model.
Type: `Cluster_ARM_Cortex-A32`.

FVP_Base_Cortex_A32x1.cluster0.cpu0

ARM Cortex-A32 CT model.
Type: `ARM_Cortex-A32`.

FVP_Base_Cortex_A32x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A32x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A32x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A32x1.cluster0.l2_cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A32x1.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A32x1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A32x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A32x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A32x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.5 FVP_Base_Cortex-A32x2

FVP_Base_Cortex-A32x2 contains the following instances:

FVP_Base_Cortex-A32x2 instances

FVP_Base_Cortex_A32x2

Base Platform Compute Subsystem for ARMCortexA32x2CT.
Type: FVP_Base_Cortex_A32x2.

FVP_Base_Cortex_A32x2.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A32x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A32x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A32x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A32x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A32x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A32x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x2.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x2.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x2.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x2.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x2.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x2.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x2.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A32x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A32x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A32x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A32x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A32x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A32x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A32x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A32x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A32x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A32x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A32x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A32x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A32x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A32x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A32x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A32x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A32x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A32x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A32x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A32x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A32x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A32x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A32x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A32x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A32x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A32x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A32x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A32x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A32x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A32x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A32x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A32x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A32x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A32x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A32x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A32x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A32x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A32x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A32x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A32x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A32x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A32x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A32x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A32x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A32x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A32x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A32x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A32x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A32x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A32x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A32x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A32x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A32x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A32x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A32x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x2.cluster0

ARM Cortex-A32 Cluster CT model.
Type: Cluster_ARM_Cortex-A32.

FVP_Base_Cortex_A32x2.cluster0.cpu0

ARM Cortex-A32 CT model.
Type: ARM_Cortex-A32.

FVP_Base_Cortex_A32x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A32x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x2.cluster0.cpu1
ARM Cortex-A32 CT model.
Type: ARM_Cortex-A32.

FVP_Base_Cortex_A32x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A32x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x2.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A32x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A32x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A32x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A32x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.6 FVP_Base_Cortex-A32x4

FVP_Base_Cortex-A32x4 contains the following instances:

FVP_Base_Cortex-A32x4 instances

FVP_Base_Cortex_A32x4

Base Platform Compute Subsystem for ARMCortexA32x4CT.
Type: `FVP_Base_Cortex_A32x4`.

FVP_Base_Cortex_A32x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A32x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A32x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A32x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A32x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A32x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A32x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A32x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A32x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A32x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A32x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A32x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A32x4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A32x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A32x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A32x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A32x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A32x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A32x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A32x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A32x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A32x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A32x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A32x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A32x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A32x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A32x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A32x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A32x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A32x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A32x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A32x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A32x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A32x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A32x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A32x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A32x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A32x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A32x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A32x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A32x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A32x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A32x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A32x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A32x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A32x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A32x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A32x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A32x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A32x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A32x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A32x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A32x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A32x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A32x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A32x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A32x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A32x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A32x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A32x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A32x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A32x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A32x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A32x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A32x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A32x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A32x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A32x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A32x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A32x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A32x4.cluster0

ARM Cortex-A32 Cluster CT model.
Type: `Cluster_ARM_Cortex-A32`.

FVP_Base_Cortex_A32x4.cluster0.cpu0

ARM Cortex-A32 CT model.
Type: `ARM_Cortex-A32`.

FVP_Base_Cortex_A32x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A32x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A32x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A32x4.cluster0.cpu1

ARM Cortex-A32 CT model.
Type: `ARM_Cortex-A32`.

FVP_Base_Cortex_A32x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A32x4.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A32x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x4.cluster0.cpu2
ARM Cortex-A32 CT model.
Type: ARM_Cortex-A32.

FVP_Base_Cortex_A32x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A32x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x4.cluster0.cpu3
ARM Cortex-A32 CT model.
Type: ARM_Cortex-A32.

FVP_Base_Cortex_A32x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A32x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x4.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A32x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A32x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A32x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A32x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.

Type: GIC_IRI.

FVP_Base_Cortex_A32x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.7 FVP_Base_Cortex-A35x1

FVP_Base_Cortex-A35x1 contains the following instances:

FVP_Base_Cortex-A35x1 instances

FVP_Base_Cortex_A35x1
Base Platform Compute Subsystem for ARMCortexA35x1CT.
Type: FVP_Base_Cortex_A35x1.

FVP_Base_Cortex_A35x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A35x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A35x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A35x1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A35x1.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A35x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A35x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A35x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A35x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A35x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A35x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A35x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A35x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A35x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A35x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A35x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A35x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A35x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A35x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A35x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A35x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A35x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A35x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A35x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A35x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A35x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A35x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A35x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A35x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A35x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A35x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A35x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A35x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A35x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A35x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A35x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A35x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A35x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A35x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A35x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A35x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A35x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A35x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A35x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A35x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A35x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A35x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A35x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A35x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A35x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A35x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A35x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A35x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A35x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A35x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A35x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A35x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A35x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A35x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A35x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A35x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A35x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A35x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A35x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A35x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x1.cluster0

ARM Cortex-A35 Cluster CT model.
Type: `Cluster_ARM_Cortex-A35`.

FVP_Base_Cortex_A35x1.cluster0.cpu0

ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A35x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A35x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A35x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A35x1.cluster0.l2_cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A35x1.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A35x1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A35x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A35x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A35x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.8 FVP_Base_Cortex-A35x2

FVP_Base_Cortex-A35x2 contains the following instances:

FVP_Base_Cortex-A35x2 instances

FVP_Base_Cortex_A35x2

Base Platform Compute Subsystem for ARMCortexA35x2CT.
Type: FVP_Base_Cortex_A35x2.

FVP_Base_Cortex_A35x2.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A35x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A35x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A35x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A35x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A35x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A35x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A35x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A35x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A35x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x2.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x2.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x2.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x2.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x2.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x2.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x2.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A35x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A35x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A35x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A35x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A35x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A35x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A35x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A35x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A35x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A35x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A35x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A35x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A35x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A35x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A35x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A35x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A35x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A35x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A35x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A35x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A35x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A35x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A35x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A35x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A35x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A35x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A35x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A35x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A35x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A35x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A35x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A35x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A35x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A35x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A35x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A35x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A35x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A35x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A35x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A35x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A35x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A35x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A35x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A35x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A35x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A35x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A35x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A35x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A35x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A35x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A35x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A35x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A35x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A35x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A35x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x2.cluster0

ARM Cortex-A35 Cluster CT model.
Type: Cluster_ARM_Cortex-A35.

FVP_Base_Cortex_A35x2.cluster0.cpu0

ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A35x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A35x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x2.cluster0.cpu1
ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A35x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A35x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x2.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A35x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A35x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A35x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A35x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.9 FVP_Base_Cortex-A35x4

FVP_Base_Cortex-A35x4 contains the following instances:

FVP_Base_Cortex-A35x4 instances

FVP_Base_Cortex_A35x4

Base Platform Compute Subsystem for ARMCortexA35x4CT.
Type: `FVP_Base_Cortex_A35x4`.

FVP_Base_Cortex_A35x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A35x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A35x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A35x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A35x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A35x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A35x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A35x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A35x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A35x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A35x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A35x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A35x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A35x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A35x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A35x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A35x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A35x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A35x4.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A35x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A35x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A35x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A35x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A35x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A35x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A35x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A35x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A35x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A35x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A35x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A35x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A35x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A35x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A35x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A35x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A35x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A35x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A35x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A35x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A35x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A35x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A35x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A35x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A35x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A35x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A35x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A35x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A35x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A35x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A35x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A35x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A35x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A35x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A35x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A35x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A35x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A35x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A35x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A35x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A35x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A35x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A35x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A35x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A35x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A35x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A35x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A35x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A35x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A35x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A35x4.cluster0

ARM Cortex-A35 Cluster CT model.
Type: `Cluster_ARM_Cortex-A35`.

FVP_Base_Cortex_A35x4.cluster0.cpu0

ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A35x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A35x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A35x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A35x4.cluster0.cpu1

ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A35x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A35x4.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A35x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x4.cluster0.cpu2
ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A35x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A35x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x4.cluster0.cpu3
ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A35x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A35x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x4.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A35x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A35x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A35x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A35x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.

Type: GIC_IRI.

FVP_Base_Cortex_A35x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.10 FVP_Base_Cortex-A510x1

FVP_Base_Cortex-A510x1 contains the following instances:

FVP_Base_Cortex-A510x1 instances

FVP_Base_Cortex_A510x1
Base Platform Compute Subsystem for ARMCortexA510x1CT.
Type: FVP_Base_Cortex_A510x1.

FVP_Base_Cortex_A510x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A510x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A510x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x1.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A510x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A510x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A510x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A510x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A510x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A510x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A510x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A510x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A510x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A510x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A510x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A510x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A510x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A510x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A510x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A510x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A510x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A510x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A510x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A510x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A510x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A510x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A510x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A510x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A510x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A510x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A510x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A510x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A510x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A510x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A510x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A510x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A510x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A510x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A510x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A510x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A510x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A510x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A510x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A510x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A510x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A510x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A510x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A510x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A510x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A510x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A510x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A510x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A510x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A510x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A510x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A510x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A510x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A510x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A510x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A510x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A510x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x1.cluster0

ARM Cortex-A510Cluster CT model.
Type: `Cluster_ARM_Cortex-A510`.

FVP_Base_Cortex_A510x1.cluster0.cpu0

ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x1.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A510x1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A510x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A510x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A510x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.11 FVP_Base_Cortex-A510x2

FVP_Base_Cortex-A510x2 contains the following instances:

FVP_Base_Cortex-A510x2 instances

FVP_Base_Cortex_A510x2

Base Platform Compute Subsystem for ARMCortexA510x2CT.
Type: FVP_Base_Cortex_A510x2.

FVP_Base_Cortex_A510x2.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A510x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A510x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A510x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A510x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A510x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x2.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x2.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x2.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x2.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x2.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x2.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x2.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A510x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A510x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A510x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A510x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A510x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A510x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A510x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A510x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A510x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A510x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A510x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A510x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A510x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A510x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A510x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A510x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A510x2.bp.pl111_clcd.pl111x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A510x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A510x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A510x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A510x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A510x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A510x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A510x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A510x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A510x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A510x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A510x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A510x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A510x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A510x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A510x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A510x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A510x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A510x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A510x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A510x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A510x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A510x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A510x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A510x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A510x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A510x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x2.cluster0

ARM Cortex-A510Cluster CT model.
Type: Cluster_ARM_Cortex-A510.

FVP_Base_Cortex_A510x2.cluster0.cpu0

ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_Cortex_A510x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A510x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x2.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x2.cluster0.cpu1
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_Cortex_A510x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A510x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A510x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A510x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A510x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A510x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.12 FVP_Base_Cortex-A510x4

FVP_Base_Cortex-A510x4 contains the following instances:

FVP_Base_Cortex-A510x4 instances

FVP_Base_Cortex_A510x4

Base Platform Compute Subsystem for ARMCortexA510x4CT.
Type: `FVP_Base_Cortex_A510x4`.

FVP_Base_Cortex_A510x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A510x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A510x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A510x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A510x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A510x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A510x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A510x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A510x4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A510x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A510x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A510x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A510x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A510x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A510x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A510x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A510x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A510x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A510x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A510x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A510x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A510x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A510x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A510x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A510x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A510x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A510x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A510x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A510x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A510x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A510x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A510x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A510x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A510x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A510x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A510x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A510x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A510x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A510x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A510x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A510x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A510x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A510x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A510x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A510x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A510x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A510x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A510x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A510x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A510x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A510x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A510x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A510x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A510x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A510x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x4.cluster0

ARM Cortex-A510Cluster CT model.
Type: `Cluster_ARM_Cortex-A510`.

FVP_Base_Cortex_A510x4.cluster0.cpu0

ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x4.cluster0.cpu1

ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu2
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_Cortex_A510x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A510x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu3
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_Cortex_A510x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A510x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x4.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A510x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A510x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A510x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A510x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.13 FVP_Base_Cortex-A510x4+Cortex-A710x4

FVP_Base_Cortex-A510x4+Cortex-A710x4 contains the following instances:

FVP_Base_Cortex-A510x4+Cortex-A710x4 instances

FVP_Base_Cortex_A510x4_Cortex_A710x4

Base Platform Compute Subsystem for ARMCortexA510x4CT_CortexA710x4CT.
Type: FVP_Base_Cortex_A510x4_Cortex_A710x4.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clock300MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clock32KHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A510x4_Cortex_A710x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0

ARM Cortex-A710 CT heterogeneous cluster model.
Type: Cluster_ARM_CortexA710_Heterogeneous_Cluster.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu0

ARM Cortex-A510 CT Model in Heterogeneous Cluster.
Type: Cluster_ARM_Cortex-A510.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu1

ARM Cortex-A510 CT Model in Heterogeneous Cluster.
Type: Cluster_ARM_Cortex-A510.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu2
ARM Cortex-A510 CT Model in Heterogeneous Cluster.
Type: `Cluster_ARM_Cortex-A510`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu3
ARM Cortex-A510 CT Model in Heterogeneous Cluster.
Type: `Cluster_ARM_Cortex-A510`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu4
ARM Cortex-A710 CT Model in Heterogeneous Cluster.
Type: `Cluster_ARM_CortexA710`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu5
ARM Cortex-A710 CT Model in Heterogeneous Cluster.
Type: `Cluster_ARM_CortexA710`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu6
ARM Cortex-A710 CT Model in Heterogeneous Cluster.
Type: `Cluster_ARM_CortexA710`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.cpu7
ARM Cortex-A710 CT Model in Heterogeneous Cluster.
Type: `Cluster_ARM_CortexA710`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.subcluster0
ARM Cortex-A510 CT Model in Heterogeneous SubCluster.
Type: `Subcluster_ARM_Cortex-A510`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0.subcluster1
ARM Cortex-A710 Cluster CT model.
Type: `Subcluster_ARM_CortexA710`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A510x4_Cortex_A710x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.14 FVP_Base_Cortex-A510x8

FVP_Base_Cortex-A510x8 contains the following instances:

FVP_Base_Cortex-A510x8 instances

FVP_Base_Cortex_A510x8

Base Platform Compute Subsystem for ARMCortexA510x8CT.
Type: `FVP_Base_Cortex_A510x8`.

FVP_Base_Cortex_A510x8.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A510x8.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A510x8.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x8.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x8.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A510x8.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x8.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A510x8.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A510x8.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A510x8.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A510x8.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A510x8.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x8.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A510x8.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A510x8.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x8.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A510x8.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x8.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A510x8.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A510x8.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A510x8.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x8.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A510x8.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A510x8.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A510x8.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A510x8.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A510x8.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A510x8.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A510x8.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A510x8.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A510x8.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A510x8.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A510x8.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x8.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x8.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A510x8.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x8.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A510x8.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A510x8.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x8.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x8.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A510x8.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x8.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A510x8.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A510x8.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A510x8.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A510x8.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A510x8.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A510x8.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A510x8.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A510x8.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A510x8.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A510x8.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A510x8.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A510x8.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A510x8.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A510x8.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A510x8.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A510x8.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A510x8.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x8.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x8.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x8.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A510x8.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A510x8.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A510x8.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A510x8.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A510x8.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A510x8.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A510x8.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A510x8.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A510x8.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A510x8.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A510x8.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A510x8.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A510x8.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A510x8.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A510x8.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A510x8.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A510x8.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A510x8.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A510x8.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A510x8.cluster0

ARM Cortex-A510Cluster CT model.
Type: `Cluster_ARM_Cortex-A510`.

FVP_Base_Cortex_A510x8.cluster0.cpu0

ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x8.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x8.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu1

ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x8.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x8.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu2
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_Cortex_A510x8.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A510x8.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu3
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_Cortex_A510x8.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A510x8.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu4
ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x8.cluster0.cpu4.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x8.cluster0.cpu4.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu4.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu4.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu5
ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x8.cluster0.cpu5.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x8.cluster0.cpu5.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu5.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu5.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu6
ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_Cortex_A510x8.cluster0.cpu6.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A510x8.cluster0.cpu6.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A510x8.cluster0.cpu6.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu6.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu7
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_Cortex_A510x8.cluster0.cpu7.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A510x8.cluster0.cpu7.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu7.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0.cpu7.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A510x8.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A510x8.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A510x8.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A510x8.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A510x8.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.15 FVP_Base_Cortex-A53x1

FVP_Base_Cortex-A53x1 contains the following instances:

FVP_Base_Cortex-A53x1 instances

FVP_Base_Cortex_A53x1

Base Platform Compute Subsystem for ARMCortexA53x1CT.
Type: `FVP_Base_Cortex_A53x1`.

FVP_Base_Cortex_A53x1.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A53x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A53x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A53x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A53x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A53x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A53x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A53x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A53x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A53x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A53x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A53x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A53x1.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A53x1.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A53x1.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A53x1.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A53x1.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A53x1.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A53x1.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A53x1.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A53x1.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A53x1.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A53x1.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A53x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A53x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A53x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A53x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A53x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A53x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A53x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A53x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A53x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A53x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A53x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A53x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A53x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A53x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A53x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A53x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A53x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A53x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A53x1.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A53x1.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A53x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A53x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A53x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A53x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A53x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A53x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A53x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A53x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A53x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A53x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A53x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A53x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A53x1.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A53x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x1.cluster0

ARM Cortex-A53 Cluster CT model.
Type: `Cluster_ARM_Cortex-A53`.

FVP_Base_Cortex_A53x1.cluster0.cpu0

ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A53x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A53x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A53x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A53x1.cluster0.l2_cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A53x1.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A53x1.dapmemlogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A53x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A53x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A53x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.16 FVP_Base_Cortex-A53x2

FVP_Base_Cortex-A53x2 contains the following instances:

FVP_Base_Cortex-A53x2 instances

FVP_Base_Cortex_A53x2
Base Platform Compute Subsystem for ARMCortexA53x2CT.
Type: FVP_Base_Cortex_A53x2.

FVP_Base_Cortex_A53x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A53x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A53x2.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A53x2.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A53x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A53x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A53x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A53x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A53x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A53x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A53x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A53x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A53x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A53x2.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A53x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A53x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A53x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A53x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A53x2.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A53x2.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A53x2.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A53x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A53x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A53x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A53x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A53x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A53x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A53x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A53x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A53x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A53x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A53x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A53x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A53x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A53x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A53x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A53x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A53x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A53x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A53x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A53x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A53x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A53x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A53x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A53x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A53x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A53x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A53x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A53x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A53x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A53x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A53x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A53x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A53x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A53x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A53x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A53x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A53x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A53x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A53x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A53x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A53x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A53x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A53x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A53x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A53x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A53x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x2.cluster0

ARM Cortex-A53 Cluster CT model.
Type: Cluster_ARM_Cortex-A53.

FVP_Base_Cortex_A53x2.cluster0.cpu0

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A53x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A53x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x2.cluster0.cpu1
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A53x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A53x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x2.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A53x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A53x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A53x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A53x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.17 FVP_Base_Cortex-A53x4

FVP_Base_Cortex-A53x4 contains the following instances:

FVP_Base_Cortex-A53x4 instances

FVP_Base_Cortex_A53x4

Base Platform Compute Subsystem for ARMCortexA53x4CT.
Type: `FVP_Base_Cortex_A53x4`.

FVP_Base_Cortex_A53x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A53x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A53x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A53x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A53x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A53x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A53x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A53x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A53x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A53x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A53x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A53x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A53x4.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A53x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A53x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A53x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A53x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A53x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A53x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A53x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A53x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A53x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A53x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A53x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A53x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A53x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A53x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A53x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A53x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A53x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A53x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A53x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A53x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A53x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A53x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A53x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A53x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A53x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A53x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A53x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A53x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A53x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A53x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A53x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A53x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A53x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A53x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A53x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A53x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A53x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A53x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A53x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A53x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A53x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A53x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A53x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A53x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A53x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A53x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A53x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A53x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A53x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A53x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A53x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A53x4.cluster0

ARM Cortex-A53 Cluster CT model.
Type: `Cluster_ARM_Cortex-A53`.

FVP_Base_Cortex_A53x4.cluster0.cpu0

ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A53x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A53x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A53x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A53x4.cluster0.cpu1

ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A53x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A53x4.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A53x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x4.cluster0.cpu2
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A53x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A53x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x4.cluster0.cpu3
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A53x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A53x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x4.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A53x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A53x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A53x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A53x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.

Type: GIC_IRI.

FVP_Base_Cortex_A53x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.18 FVP_Base_Cortex-A55

FVP_Base_Cortex-A55 contains the following instances:

FVP_Base_Cortex-A55 instances

FVP_Base_Cortex_A55
Base Platform Compute Subsystem for ARMCortexA55CT.
Type: FVP_Base_Cortex_A55.

FVP_Base_Cortex_A55.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A55.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A55.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A55.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A55.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A55.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A55.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A55.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A55.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A55.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A55.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A55.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A55.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A55.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A55.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A55.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A55.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A55.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A55.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A55.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A55.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A55.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A55.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A55.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A55.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A55.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A55.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A55.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55.cluster0

ARM Cortex-A55 Cluster CT model.
Type: `Cluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A55.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu1
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu2
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu3
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu4
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55.cluster0.cpu4.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55.cluster0.cpu4.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu4.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu4.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu5
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55.cluster0.cpu5.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55.cluster0.cpu5.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu5.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu5.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55.cluster0.cpu6
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55.cluster0.cpu6.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A55.cluster0.cpu6.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu6.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu6.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu7
ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55.cluster0.cpu7.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A55.cluster0.cpu7.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu7.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0.cpu7.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55.pctl
Base Platforms Power Controller.

Type: `Base_PowerController`.

7.19 FVP_Base_Cortex-A55+Cortex-A76

FVP_Base_Cortex-A55+Cortex-A76 contains the following instances:

FVP_Base_Cortex-A55+Cortex-A76 instances

`FVP_Base_Cortex_A55_Cortex_A76`

Base Platform Compute Subsystem for ARMCortexA55CT_CortexA76CT.
Type: `FVP_Base_Cortex_A55_Cortex_A76`.

`FVP_Base_Cortex_A55_Cortex_A76.bp`

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55_Cortex_A76.bp.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55_Cortex_A76.bp.ap_refclk
ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55_Cortex_A76.bp.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55_Cortex_A76.bp.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55_Cortex_A76.bp.clock24MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55_Cortex_A76.bp.clock300MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55_Cortex_A76.bp.clock32KHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55_Cortex_A76.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55_Cortex_A76.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55_Cortex_A76.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55_Cortex_A76.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55_Cortex_A76.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55_Cortex_A76.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55_Cortex_A76.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55_Cortex_A76.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55_Cortex_A76.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55_Cortex_A76.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55_Cortex_A76.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55_Cortex_A76.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a

proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart2`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart2.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart3`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl011_uart3.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A55_Cortex_A76.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl111_clcd
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl111_clcd.pl11x_clcd
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer.timer.thread
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl111_clcd_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55_Cortex_A76.bp.pl180_mci
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A55_Cortex_A76.bp.ps2keyboard
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A55_Cortex_A76.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A55_Cortex_A76.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55_Cortex_A76.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A55_Cortex_A76.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A55_Cortex_A76.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55_Cortex_A76.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55_Cortex_A76.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55_Cortex_A76.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55_Cortex_A76.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A55_Cortex_A76.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A55_Cortex_A76.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A55_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55_Cortex_A76.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55_Cortex_A76.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55_Cortex_A76.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55_Cortex_A76.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55_Cortex_A76.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55_Cortex_A76.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A55_Cortex_A76.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A55_Cortex_A76.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A55_Cortex_A76.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55_Cortex_A76.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55_Cortex_A76.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55_Cortex_A76.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55_Cortex_A76.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55_Cortex_A76.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55_Cortex_A76.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55_Cortex_A76.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55_Cortex_A76.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55_Cortex_A76.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A55_Cortex_A76.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55_Cortex_A76.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55_Cortex_A76.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55_Cortex_A76.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55_Cortex_A76.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55_Cortex_A76.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A55_Cortex_A76.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55_Cortex_A76.cluster0

ARM Cortex-A55_Cortex-A76 Cluster CT model.
Type: `Cluster_ARM_Cortex-A55_Cortex-A76`.

FVP_Base_Cortex_A55_Cortex_A76.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55_Cortex_A76.cluster0.cpu1

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55_Cortex_A76.cluster0.subcluster0

ARM Cortex-A55 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55_Cortex_A76.cluster0.subcluster1

ARM Cortex-A76 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A76`.

FVP_Base_Cortex_A55_Cortex_A76.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A55_Cortex_A76.dapmemlogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55_Cortex_A76.elfloader

ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55_Cortex_A76.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A55_Cortex_A76.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.20 FVP_Base_Cortex-A55x1

FVP_Base_Cortex-A55x1 contains the following instances:

FVP_Base_Cortex-A55x1 instances

FVP_Base_Cortex_A55x1
Base Platform Compute Subsystem for ARMCortexA55x1CT.
Type: FVP_Base_Cortex_A55x1.

FVP_Base_Cortex_A55x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A55x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1.bp.Timer_2_3
ARM Dual-Timer Module(SP804).

Type: SP804_Timer.

FVP_Base_Cortex_A55x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A55x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A55x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A55x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A55x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A55x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A55x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A55x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A55x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A55x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A55x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A55x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A55x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A55x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A55x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A55x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A55x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A55x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A55x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A55x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A55x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A55x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A55x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A55x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A55x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x1.cluster0

ARM Cortex-A55 Cluster CT model.
Type: `Cluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55x1.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A55x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x1.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A55x1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A55x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A55x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A55x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.21 FVP_Base_Cortex-A55x1+Cortex-A75x1

FVP_Base_Cortex-A55x1+Cortex-A75x1 contains the following instances:

FVP_Base_Cortex-A55x1+Cortex-A75x1 instances

FVP_Base_Cortex_A55x1_Cortex_A75x1

Base Platform Compute Subsystem for ARMCortexA55x1CT_CortexA75x1CT.
Type: FVP_Base_Cortex_A55x1_Cortex_A75x1.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clock300MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clock32KHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x1_Cortex_A75x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A55x1_Cortex_A75x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x1_Cortex_A75x1.cluster0

ARM Cortex-A55_Cortex-A75 Cluster CT model.
Type: Cluster_ARM_Cortex-A55_Cortex-A75.

FVP_Base_Cortex_A55x1_Cortex_A75x1.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x1_Cortex_A75x1.cluster0.cpu1

ARM Cortex-A75 CT model.
Type: ARM_Cortex-A75.

FVP_Base_Cortex_A55x1_Cortex_A75x1.cluster0.subcluster0
ARM Cortex-A55 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55x1_Cortex_A75x1.cluster0.subcluster1
ARM Cortex-A75 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A75`.

FVP_Base_Cortex_A55x1_Cortex_A75x1.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x1_Cortex_A75x1.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55x1_Cortex_A75x1.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55x1_Cortex_A75x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55x1_Cortex_A75x1.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.22 FVP_Base_Cortex-A55x2

FVP_Base_Cortex-A55x2 contains the following instances:

FVP_Base_Cortex-A55x2 instances

FVP_Base_Cortex_A55x2
Base Platform Compute Subsystem for ARMCortexA55x2CT.
Type: `FVP_Base_Cortex_A55x2`.

FVP_Base_Cortex_A55x2.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A55x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x2.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A55x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A55x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A55x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A55x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A55x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A55x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A55x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A55x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A55x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Cortex_A55x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A55x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A55x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A55x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A55x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A55x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A55x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A55x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A55x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A55x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A55x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A55x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A55x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A55x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2.cluster0

ARM Cortex-A55 Cluster CT model.
Type: Cluster_ARM_Cortex-A55.

FVP_Base_Cortex_A55x2.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A55x2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x2.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x2.cluster0.cpu1
ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A55x2.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x2.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x2.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55x2.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55x2.pctl
Base Platforms Power Controller.

Type: `Base_PowerController`.

7.23 FVP_Base_Cortex-A55x2+Cortex-A75x2

FVP_Base_Cortex-A55x2+Cortex-A75x2 contains the following instances:

FVP_Base_Cortex-A55x2+Cortex-A75x2 instances

`FVP_Base_Cortex_A55x2_Cortex_A75x2`

Base Platform Compute Subsystem for ARMCortexA55x2CT_CortexA75x2CT.
Type: `FVP_Base_Cortex_A55x2_Cortex_A75x2`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp`

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.ap_refclk
ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clock24MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clock300MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clock32KHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a

proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart2`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart2.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart3`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl011_uart3.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl111_clcd
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl111_clcd_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.pl180_mci
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.ps2keyboard
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0

ARM Cortex-A55_Cortex-A75 Cluster CT model.
Type: `Cluster_ARM_Cortex-A55_Cortex-A75`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0.cpu1

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0.cpu2

ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0.cpu3

ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0.subcluster0

ARM Cortex-A55 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0.subcluster1

ARM Cortex-A75 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A75`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A55x2_Cortex_A75x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A55x2_Cortex_A75x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A55x2_Cortex_A75x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A55x2_Cortex_A75x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.24 FVP_Base_Cortex-A55x4

FVP_Base_Cortex-A55x4 contains the following instances:

FVP_Base_Cortex-A55x4 instances

FVP_Base_Cortex_A55x4
Base Platform Compute Subsystem for ARMCortexA55x4CT.
Type: FVP_Base_Cortex_A55x4.

FVP_Base_Cortex_A55x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A55x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A55x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A55x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A55x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A55x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A55x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A55x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A55x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A55x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A55x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A55x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A55x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A55x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A55x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A55x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A55x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A55x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A55x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A55x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A55x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A55x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A55x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A55x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A55x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A55x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A55x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4.cluster0

ARM Cortex-A55 Cluster CT model.
Type: Cluster_ARM_Cortex-A55.

FVP_Base_Cortex_A55x4.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55x4.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu1
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu2
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A55x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A55x4.cluster0.cpu3
ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A55x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x4.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A55x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55x4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.25 FVP_Base_Cortex-A55x4+Cortex-A75x1

FVP_Base_Cortex-A55x4+Cortex-A75x1 contains the following instances:

FVP_Base_Cortex-A55x4+Cortex-A75x1 instances

FVP_Base_Cortex_A55x4_Cortex_A75x1
Base Platform Compute Subsystem for ARMCortexA55x4CT_CortexA75x1CT.
Type: `FVP_Base_Cortex_A55x4_Cortex_A75x1`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp
Peripherals and address map for the Base Platform.

Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_2_3.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_2_3.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.Timer_2_3.counter1`

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl111_clcd_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.refcounter`
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.reset_or`
Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.sram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.trusted_key_storage`
Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.trusted_nv_counter`
Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.trusted_rng`
Random Number Generator unit.
Type: `RandomNumberGenerator`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.trusted_watchdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.tzc_400`
TrustZone Address Space Controller.
Type: `TZC_400`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.vedcc`
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

`FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.virtio_net`
VirtioNet device over MMIO transport.
Type: `VirtioNetMMIO`.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.bp.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cci400

Cache Coherent Interconnect for AXI4 ACE.

Type: CCI400.

FVP_Base_Cortex_A55x4_Cortex_A75x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0
ARM Cortex-A55_Cortex-A75 Cluster CT model.
Type: Cluster_ARM_Cortex-A55_Cortex-A75.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0.cpu0
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0.cpu1
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0.cpu2
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0.cpu3
ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0.cpu4
ARM Cortex-A75 CT model.
Type: ARM_Cortex-A75.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0.subcluster0
ARM Cortex-A55 Cluster CT model.
Type: Subcluster_ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0.subcluster1
ARM Cortex-A75 Cluster CT model.
Type: Subcluster_ARM_Cortex-A75.

FVP_Base_Cortex_A55x4_Cortex_A75x1.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A55x4_Cortex_A75x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A55x4_Cortex_A75x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.26 FVP_Base_Cortex-A55x4+Cortex-A75x2

FVP_Base_Cortex-A55x4+Cortex-A75x2 contains the following instances:

FVP_Base_Cortex-A55x4+Cortex-A75x2 instances

FVP_Base_Cortex_A55x4_Cortex_A75x2

Base Platform Compute Subsystem for ARMCortexA55x4CT_CortexA75x2CT.
Type: `FVP_Base_Cortex_A55x4_Cortex_A75x2`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0

ARM Cortex-A55_Cortex-A75 Cluster CT model.
Type: `Cluster_ARM_Cortex-A55_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.cpu1

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.cpu2

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.cpu3

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.cpu4

ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.cpu5

ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.subcluster0

ARM Cortex-A55 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0.subcluster1
ARM Cortex-A75 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55x4_Cortex_A75x2.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.27 FVP_Base_Cortex-A55x4+Cortex-A75x4

FVP_Base_Cortex-A55x4+Cortex-A75x4 contains the following instances:

FVP_Base_Cortex-A55x4+Cortex-A75x4 instances

FVP_Base_Cortex_A55x4_Cortex_A75x4
Base Platform Compute Subsystem for ARMCortexA55x4CT_CortexA75x4CT.
Type: `FVP_Base_Cortex_A55x4_Cortex_A75x4`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.ap_refclk
ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dmc

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A55x4_Cortex_A75x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0

ARM Cortex-A55_Cortex-A75 Cluster CT model.
Type: Cluster_ARM_Cortex-A55_Cortex-A75.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu1

ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu2
ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu3
ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu4
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu5
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu6
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.cpu7
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.subcluster0
ARM Cortex-A55 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0.subcluster1
ARM Cortex-A75 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A75`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55x4_Cortex_A75x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.28 FVP_Base_Cortex-A55x4+Cortex-A76x2

FVP_Base_Cortex-A55x4+Cortex-A76x2 contains the following instances:

FVP_Base_Cortex-A55x4+Cortex-A76x2 instances

FVP_Base_Cortex_A55x4_Cortex_A76x2

Base Platform Compute Subsystem for ARMCortexA55x4CT_CortexA76x2CT.
Type: `FVP_Base_Cortex_A55x4_Cortex_A76x2`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0

ARM Cortex-A55_Cortex-A76 Cluster CT model.
Type: `Cluster_ARM_Cortex-A55_Cortex-A76`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.cpu1

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.cpu2

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.cpu3

ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.cpu4

ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.cpu5

ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.subcluster0

ARM Cortex-A55 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0.subcluster1
ARM Cortex-A76 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A76`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55x4_Cortex_A76x2.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.29 FVP_Base_Cortex-A55x4+Cortex-A78x4

FVP_Base_Cortex-A55x4+Cortex-A78x4 contains the following instances:

FVP_Base_Cortex-A55x4+Cortex-A78x4 instances

FVP_Base_Cortex_A55x4_Cortex_A78x4
Base Platform Compute Subsystem for ARMCortexA55x4CT_CortexA78x4CT.
Type: `FVP_Base_Cortex_A55x4_Cortex_A78x4`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.ap_refclk
ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dmc

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A55x4_Cortex_A78x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0

ARM Cortex-A55_Cortex-A78 Cluster CT model.
Type: Cluster_ARM_Cortex-A55_Cortex-A78.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu0

ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu1

ARM Cortex-A55 CT model.
Type: ARM_Cortex-A55.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu2
ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu3
ARM Cortex-A55 CT model.
Type: `ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu4
ARM Cortex-A78 CT model.
Type: `ARM_Cortex-A78`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu5
ARM Cortex-A78 CT model.
Type: `ARM_Cortex-A78`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu6
ARM Cortex-A78 CT model.
Type: `ARM_Cortex-A78`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.cpu7
ARM Cortex-A78 CT model.
Type: `ARM_Cortex-A78`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.subcluster0
ARM Cortex-A55 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A55`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0.subcluster1
ARM Cortex-A78 Cluster CT model.
Type: `Subcluster_ARM_Cortex-A78`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A55x4_Cortex_A78x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.30 FVP_Base_Cortex-A57x1

FVP_Base_Cortex-A57x1 contains the following instances:

FVP_Base_Cortex-A57x1 instances

FVP_Base_Cortex_A57x1

Base Platform Compute Subsystem for ARMCortexA57x1CT.
Type: `FVP_Base_Cortex_A57x1`.

FVP_Base_Cortex_A57x1.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A57x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A57x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A57x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A57x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A57x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A57x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A57x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A57x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A57x1.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A57x1.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A57x1.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A57x1.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A57x1.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A57x1.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A57x1.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A57x1.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A57x1.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x1.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A57x1.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A57x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A57x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A57x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A57x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A57x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A57x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A57x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A57x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A57x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A57x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A57x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A57x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A57x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A57x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A57x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A57x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A57x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A57x1.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A57x1.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A57x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A57x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A57x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A57x1.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A57x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1.cluster0

ARM Cortex-A57 Cluster CT model.
Type: `Cluster_ARM_Cortex-A57`.

FVP_Base_Cortex_A57x1.cluster0.cpu0

ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1.cluster0.l2_cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x1.dapmemlogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A57x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.31 FVP_Base_Cortex-A57x1-A35x1

FVP_Base_Cortex-A57x1-A35x1 contains the following instances:

FVP_Base_Cortex-A57x1-A35x1 instances

FVP_Base_Cortex_A57x1_A35x1
Base Platform Compute Subsystem for ARMCortexA57x1CT and ARMCortexA35x1CT.
Type: FVP_Base_Cortex_A57x1_A35x1.

FVP_Base_Cortex_A57x1_A35x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x1_A35x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x1_A35x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A57x1_A35x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A57x1_A35x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A35x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A35x1.bp.dram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A35x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x1_A35x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x1_A35x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A35x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A35x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A35x1.bp.exclusive_monitor
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A57x1_A35x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x1_A35x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x1_A35x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x1_A35x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x1_A35x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_Base_Cortex_A57x1_A35x1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Cortex_A57x1_A35x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x1_A35x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x1_A35x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x1_A35x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x1_A35x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x1_A35x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A57x1_A35x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A57x1_A35x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x1_A35x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x1_A35x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x1_A35x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A57x1_A35x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A57x1_A35x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A57x1_A35x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A57x1_A35x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A35x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A57x1_A35x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A57x1_A35x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A35x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A35x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x1_A35x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x1_A35x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A57x1_A35x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x1_A35x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A57x1_A35x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A35x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A35x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A35x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A35x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A35x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A35x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A57x1_A35x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A57x1_A35x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A57x1_A35x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x1_A35x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x1_A35x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x1_A35x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x1_A35x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x1_A35x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x1_A35x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x1_A35x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x1_A35x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x1_A35x1.bp.virtiop9device_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x1_A35x1.bp.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_Base_Cortex_A57x1_A35x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_Base_Cortex_A57x1_A35x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A35x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A57x1_A35x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A35x1.cluster0

ARM Cortex-A57 Cluster CT model.
Type: `Cluster_ARM_Cortex-A57`.

FVP_Base_Cortex_A57x1_A35x1.cluster0.cpu0
ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x1_A35x1.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x1_A35x1.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A35x1.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A35x1.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A35x1.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A57x1_A35x1.cluster1
ARM Cortex-A35 Cluster CT model.
Type: `Cluster_ARM_Cortex-A35`.

FVP_Base_Cortex_A57x1_A35x1.cluster1.cpu0
ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A57x1_A35x1.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x1_A35x1.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A35x1.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A35x1.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A35x1.cluster1_labeller
Type: `Labeller`.

FVP_Base_Cortex_A57x1_A35x1.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A57x1_A35x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x1_A35x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x1_A35x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.32 FVP_Base_Cortex-A57x1-A53x1

FVP_Base_Cortex-A57x1-A53x1 contains the following instances:

FVP_Base_Cortex-A57x1-A53x1 instances

FVP_Base_Cortex_A57x1_A53x1
Base Platform Compute Subsystem for ARMCortexA57x1CT and ARMCortexA53x1CT.
Type: FVP_Base_Cortex_A57x1_A53x1.

FVP_Base_Cortex_A57x1_A53x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x1_A53x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x1_A53x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A57x1_A53x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A57x1_A53x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A53x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A53x1.bp.dram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A53x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x1_A53x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x1_A53x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A53x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A53x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A53x1.bp.exclusive_monitor
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A57x1_A53x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x1_A53x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x1_A53x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x1_A53x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x1_A53x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_Base_Cortex_A57x1_A53x1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Cortex_A57x1_A53x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x1_A53x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x1_A53x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x1_A53x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x1_A53x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x1_A53x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A57x1_A53x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A57x1_A53x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x1_A53x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x1_A53x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A57x1_A53x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A57x1_A53x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A57x1_A53x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A57x1_A53x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A53x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A57x1_A53x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A57x1_A53x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A53x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A53x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x1_A53x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x1_A53x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A57x1_A53x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x1_A53x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A57x1_A53x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x1_A53x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x1_A53x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A53x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A53x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A53x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x1_A53x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A57x1_A53x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A57x1_A53x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A57x1_A53x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x1_A53x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x1_A53x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x1_A53x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x1_A53x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x1_A53x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x1_A53x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x1_A53x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x1_A53x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x1_A53x1.bp.virtiop9device_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x1_A53x1.bp.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_Base_Cortex_A57x1_A53x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_Base_Cortex_A57x1_A53x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x1_A53x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A57x1_A53x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x1_A53x1.cluster0

ARM Cortex-A57 Cluster CT model.
Type: `Cluster_ARM_Cortex-A57`.

FVP_Base_Cortex_A57x1_A53x1.cluster0.cpu0
ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x1_A53x1.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x1_A53x1.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A53x1.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A53x1.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A53x1.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A57x1_A53x1.cluster1
ARM Cortex-A53 Cluster CT model.
Type: `Cluster_ARM_Cortex-A53`.

FVP_Base_Cortex_A57x1_A53x1.cluster1.cpu0
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A57x1_A53x1.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x1_A53x1.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A53x1.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A53x1.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x1_A53x1.cluster1_labeller
Type: `Labeller`.

FVP_Base_Cortex_A57x1_A53x1.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A57x1_A53x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x1_A53x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x1_A53x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.33 FVP_Base_Cortex-A57x2

FVP_Base_Cortex-A57x2 contains the following instances:

FVP_Base_Cortex-A57x2 instances

FVP_Base_Cortex_A57x2
Base Platform Compute Subsystem for ARMCortexA57x2CT.
Type: FVP_Base_Cortex_A57x2.

FVP_Base_Cortex_A57x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A57x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x2.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x2.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A57x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A57x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A57x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A57x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A57x2.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A57x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x2.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x2.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A57x2.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A57x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A57x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A57x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A57x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A57x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A57x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A57x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A57x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A57x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A57x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A57x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A57x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A57x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A57x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A57x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A57x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A57x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A57x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A57x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A57x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2.cluster0

ARM Cortex-A57 Cluster CT model.
Type: Cluster_ARM_Cortex-A57.

FVP_Base_Cortex_A57x2.cluster0.cpu0

ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2.cluster0.cpu1
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A57x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.34 FVP_Base_Cortex-A57x2-A35x4

FVP_Base_Cortex-A57x2-A35x4 contains the following instances:

FVP_Base_Cortex-A57x2-A35x4 instances

FVP_Base_Cortex_A57x2_A35x4
Base Platform Compute Subsystem for ARMCortexA57x2CT and ARMCortexA35x4CT.
Type: `FVP_Base_Cortex_A57x2_A35x4`.

FVP_Base_Cortex_A57x2_A35x4.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A35x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A35x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A57x2_A35x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A57x2_A35x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x2_A35x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x2_A35x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2_A35x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2_A35x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A57x2_A35x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x2_A35x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x2_A35x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x2_A35x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x2_A35x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A57x2_A35x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A57x2_A35x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x2_A35x4.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A57x2_A35x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A57x2_A35x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A57x2_A35x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A57x2_A35x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A57x2_A35x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A57x2_A35x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A57x2_A35x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x2_A35x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2_A35x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A57x2_A35x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A57x2_A35x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A57x2_A35x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2_A35x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A57x2_A35x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2_A35x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A57x2_A35x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A57x2_A35x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x2_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A57x2_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A57x2_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x2_A35x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A57x2_A35x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A57x2_A35x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A57x2_A35x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A57x2_A35x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A57x2_A35x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A57x2_A35x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x2_A35x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x2_A35x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A57x2_A35x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A57x2_A35x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A57x2_A35x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A35x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A35x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A35x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A35x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A57x2_A35x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A57x2_A35x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A57x2_A35x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x2_A35x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x2_A35x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x2_A35x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x2_A35x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x2_A35x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2_A35x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x2_A35x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2_A35x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x2_A35x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2_A35x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A57x2_A35x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A57x2_A35x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x2_A35x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A35x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A57x2_A35x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A35x4.cluster0

ARM Cortex-A57 Cluster CT model.
Type: `Cluster_ARM_Cortex-A57`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu0

ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu1

ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A57x2_A35x4.cluster1
ARM Cortex-A35 Cluster CT model.
Type: `Cluster_ARM_Cortex-A35`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu0
ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu1
ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu1.l1icache
PV Cache.

Type: PVCache.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu2

ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu2.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu2.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu3

ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu3.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A35x4.cluster1.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A35x4.cluster1.l2_cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A35x4.cluster1_labeller

Type: Labeller.

FVP_Base_Cortex_A57x2_A35x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A57x2_A35x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x2_A35x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x2_A35x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.35 FVP_Base_Cortex-A57x2-A53x4

FVP_Base_Cortex-A57x2-A53x4 contains the following instances:

FVP_Base_Cortex-A57x2-A53x4 instances

FVP_Base_Cortex_A57x2_A53x4
Base Platform Compute Subsystem for ARMCortexA57x2CT and ARMCortexA53x4CT.
Type: `FVP_Base_Cortex_A57x2_A53x4`.

FVP_Base_Cortex_A57x2_A53x4.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A53x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x2_A53x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A57x2_A53x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A57x2_A53x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x2_A53x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x2_A53x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2_A53x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2_A53x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x2_A53x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A57x2_A53x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x2_A53x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x2_A53x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x2_A53x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x2_A53x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A57x2_A53x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A57x2_A53x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x2_A53x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x2_A53x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x2_A53x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x2_A53x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x2_A53x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A57x2_A53x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A57x2_A53x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x2_A53x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x2_A53x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A57x2_A53x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A57x2_A53x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A57x2_A53x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A57x2_A53x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A57x2_A53x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x2_A53x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x2_A53x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A57x2_A53x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A57x2_A53x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A57x2_A53x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A53x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A53x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A53x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x2_A53x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A57x2_A53x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A57x2_A53x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A57x2_A53x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x2_A53x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x2_A53x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x2_A53x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x2_A53x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x2_A53x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2_A53x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x2_A53x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2_A53x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x2_A53x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2_A53x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A57x2_A53x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A57x2_A53x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x2_A53x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A57x2_A53x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x2_A53x4.cluster0

ARM Cortex-A57 Cluster CT model.
Type: `Cluster_ARM_Cortex-A57`.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu0

ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu1
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A53x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A53x4.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A53x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A57x2_A53x4.cluster1
ARM Cortex-A53 Cluster CT model.
Type: Cluster_ARM_Cortex-A53.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu0
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu1
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu1.dtlb
TLB - instruction, data or unified.

Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu2
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu3
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x2_A53x4.cluster1_labeller
Type: `Labeller`.

FVP_Base_Cortex_A57x2_A53x4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A57x2_A53x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x2_A53x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x2_A53x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.36 FVP_Base_Cortex-A57x4

FVP_Base_Cortex-A57x4 contains the following instances:

FVP_Base_Cortex-A57x4 instances

FVP_Base_Cortex_A57x4
Base Platform Compute Subsystem for ARMCortexA57x4CT.
Type: FVP_Base_Cortex_A57x4.

FVP_Base_Cortex_A57x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A57x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A57x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A57x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A57x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A57x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A57x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A57x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A57x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A57x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A57x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A57x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A57x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A57x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A57x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A57x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A57x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A57x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A57x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A57x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A57x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A57x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A57x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A57x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A57x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A57x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A57x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A57x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A57x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4.cluster0

ARM Cortex-A57 Cluster CT model.
Type: Cluster_ARM_Cortex-A57.

FVP_Base_Cortex_A57x4.cluster0.cpu0

ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.cpu1
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.cpu2
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.cpu3
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A57x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.37 FVP_Base_Cortex-A57x4-A35x4

FVP_Base_Cortex-A57x4-A35x4 contains the following instances:

FVP_Base_Cortex-A57x4-A35x4 instances

FVP_Base_Cortex_A57x4_A35x4
Base Platform Compute Subsystem for ARMCortexA57x4CT and ARMCortexA35x4CT.
Type: FVP_Base_Cortex_A57x4_A35x4.

FVP_Base_Cortex_A57x4_A35x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x4_A35x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A57x4_A35x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A57x4_A35x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A57x4_A35x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x4_A35x4.bp.clock24MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x4_A35x4.bp.clock300MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x4_A35x4.bp.clock32KHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x4_A35x4.bp.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x4_A35x4.bp.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x4_A35x4.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A57x4_A35x4.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x4_A35x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x4_A35x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A57x4_A35x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4_A35x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4_A35x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4_A35x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4_A35x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A57x4_A35x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A57x4_A35x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x4_A35x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A57x4_A35x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A57x4_A35x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x4_A35x4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A57x4_A35x4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A57x4_A35x4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A57x4_A35x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4_A35x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A57x4_A35x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A57x4_A35x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A57x4_A35x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A57x4_A35x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A57x4_A35x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A57x4_A35x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A57x4_A35x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x4_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Cortex_A57x4_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A57x4_A35x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A57x4_A35x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A57x4_A35x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A57x4_A35x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A57x4_A35x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A57x4_A35x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A57x4_A35x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A57x4_A35x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4_A35x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4_A35x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A57x4_A35x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x4_A35x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A57x4_A35x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4_A35x4.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4_A35x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4_A35x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A57x4_A35x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A57x4_A35x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A57x4_A35x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A57x4_A35x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x4_A35x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x4_A35x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x4_A35x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x4_A35x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x4_A35x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4_A35x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x4_A35x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4_A35x4.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x4_A35x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A57x4_A35x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A57x4_A35x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A57x4_A35x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A35x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A57x4_A35x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A35x4.cluster0
ARM Cortex-A57 Cluster CT model.
Type: `Cluster_ARM_Cortex-A57`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu0
ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu1
ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu2
ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu3
ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A57x4_A35x4.cluster1
ARM Cortex-A35 Cluster CT model.
Type: `Cluster_ARM_Cortex-A35`.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu0
ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu1
ARM Cortex-A35 CT model.
Type: `ARM_Cortex-A35`.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu1.l1dcache
PV Cache.

Type: PVCache.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu1.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu2

ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu2.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu2.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu3

ARM Cortex-A35 CT model.
Type: ARM_Cortex-A35.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu3.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A35x4.cluster1.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A35x4.cluster1.l2_cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A35x4.cluster1_labeller

Type: Labeller.

FVP_Base_Cortex_A57x4_A35x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A57x4_A35x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x4_A35x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x4_A35x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.38 FVP_Base_Cortex-A57x4-A53x4

FVP_Base_Cortex-A57x4-A53x4 contains the following instances:

FVP_Base_Cortex-A57x4-A53x4 instances

FVP_Base_Cortex_A57x4_A53x4
Base Platform Compute Subsystem for ARMCortexA57x4CT and ARMCortexA53x4CT.
Type: FVP_Base_Cortex_A57x4_A53x4.

FVP_Base_Cortex_A57x4_A53x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_2_3
ARM Dual-Timer Module(SP804).

Type: SP804_Timer.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x4_A53x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A57x4_A53x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A57x4_A53x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A57x4_A53x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A53x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A53x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A57x4_A53x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A57x4_A53x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A57x4_A53x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A53x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A53x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A57x4_A53x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A57x4_A53x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4_A53x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A57x4_A53x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4_A53x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A57x4_A53x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A57x4_A53x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A57x4_A53x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A57x4_A53x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x4_A53x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x4_A53x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x4_A53x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x4_A53x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A57x4_A53x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A57x4_A53x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x4_A53x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A57x4_A53x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A57x4_A53x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A57x4_A53x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A57x4_A53x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A57x4_A53x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A57x4_A53x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A57x4_A53x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A57x4_A53x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A57x4_A53x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A57x4_A53x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A57x4_A53x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x4_A53x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x4_A53x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x4_A53x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A57x4_A53x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A57x4_A53x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A57x4_A53x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A57x4_A53x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A57x4_A53x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A57x4_A53x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A57x4_A53x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A57x4_A53x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A57x4_A53x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4_A53x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A57x4_A53x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4_A53x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A57x4_A53x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4_A53x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A57x4_A53x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A57x4_A53x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A57x4_A53x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A57x4_A53x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A57x4_A53x4.cluster0

ARM Cortex-A57 Cluster CT model.
Type: `Cluster_ARM_Cortex-A57`.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu0

ARM Cortex-A57 CT model.
Type: `ARM_Cortex-A57`.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu1
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu2
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu3
ARM Cortex-A57 CT model.
Type: ARM_Cortex-A57.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A57x4_A53x4.cluster1
ARM Cortex-A53 Cluster CT model.
Type: Cluster_ARM_Cortex-A53.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu0
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu1
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu2
ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu2.l1dcache
PV Cache.

Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu3

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu3.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1.l2_cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A57x4_A53x4.cluster1_labeller

Type: Labeller.

FVP_Base_Cortex_A57x4_A53x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A57x4_A53x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A57x4_A53x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A57x4_A53x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.39 FVP_Base_Cortex-A65AEx2

FVP_Base_Cortex-A65AEx2 contains the following instances:

FVP_Base_Cortex-A65AEx2 instances

FVP_Base_Cortex_A65AEx2

Base Platform Compute Subsystem for ARMCortexA65AEx2CT.
Type: `FVP_Base_Cortex_A65AEx2`.

FVP_Base_Cortex_A65AEx2.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A65AEx2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A65AEx2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A65AEx2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A65AEx2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A65AEx2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx2.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A65AEx2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65AEx2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65AEx2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65AEx2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A65AEx2.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A65AEx2.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A65AEx2.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A65AEx2.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A65AEx2.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A65AEx2.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A65AEx2.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A65AEx2.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A65AEx2.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx2.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A65AEx2.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A65AEx2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A65AEx2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A65AEx2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A65AEx2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A65AEx2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A65AEx2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A65AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A65AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A65AEx2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A65AEx2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A65AEx2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A65AEx2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A65AEx2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A65AEx2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A65AEx2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65AEx2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A65AEx2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A65AEx2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A65AEx2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A65AEx2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65AEx2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A65AEx2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A65AEx2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A65AEx2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A65AEx2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A65AEx2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A65AEx2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A65AEx2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A65AEx2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A65AEx2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2.cluster0

ARM Cortex-A65AE Cluster CT model.
Type: `Cluster_ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu0.thread0

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu0.thread1

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx2.cluster0.cpu1.thread0
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx2.cluster0.cpu1.thread1
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A65AEx2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A65AEx2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A65AEx2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.40 FVP_Base_Cortex-A65AEx2+Cortex-A76AEx2

FVP_Base_Cortex-A65AEx2+Cortex-A76AEx2 contains the following instances:

FVP_Base_Cortex-A65AEx2+Cortex-A76AEx2 instances

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2
Base Platform Compute Subsystem for ARMCortexA65AEx2CT_CortexA76AEx2CT.
Type: FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp
Peripherals and address map for the Base Platform.

Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_2_3.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_2_3.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.Timer_2_3.counter1`

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl111_clcd_labeller
Type: `Labeller`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.pl180_mci
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.ps2keyboard
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.ps2mouse
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.psram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.refcounter
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.reset_or
Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.sram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.terminal_0
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.terminal_1
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.terminal_2
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.terminal_3
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.trusted_rng
Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.tzc_400
TrustZone Address Space Controller.
Type: `TZC_400`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.ve_sysregs
Type: `VE_SysRegs`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: `VirtioNetMMIO`.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.bp.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cci400

Cache Coherent Interconnect for AXI4 ACE.

Type: CCI400.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0
ARM Cortex-A65AE_Cortex-A76AE Cluster CT model.
Type: Cluster_ARM_Cortex-A65AE_Cortex-A76AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.cpu0.thread0
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.cpu0.thread1
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.cpu1.thread0
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.cpu1.thread1
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.cpu2
ARM Cortex-A76AE CT model.
Type: ARM_Cortex-A76AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.cpu3
ARM Cortex-A76AE CT model.
Type: ARM_Cortex-A76AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.subcluster0
ARM Cortex-A65AE Cluster CT model.
Type: Subcluster_ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0.subcluster1
ARM Cortex-A76AE Cluster CT model.
Type: Subcluster_ARM_Cortex-A76AE.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A65AEx2_Cortex_A76AEx2.pctl

Base Platforms Power Controller.
Type: `Base_PowerController`.

7.41 FVP_Base_Cortex-A65AEx4

FVP_Base_Cortex-A65AEx4 contains the following instances:

FVP_Base_Cortex-A65AEx4 instances

FVP_Base_Cortex_A65AEx4

Base Platform Compute Subsystem for ARMCortexA65AEx4CT.
Type: `FVP_Base_Cortex_A65AEx4`.

FVP_Base_Cortex_A65AEx4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A65AEx4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A65AEx4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A65AEx4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A65AEx4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A65AEx4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A65AEx4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65AEx4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65AEx4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65AEx4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65AEx4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65AEx4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65AEx4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A65AEx4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A65AEx4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A65AEx4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A65AEx4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65AEx4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A65AEx4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A65AEx4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65AEx4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65AEx4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A65AEx4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A65AEx4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A65AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65AEx4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A65AEx4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A65AEx4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A65AEx4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A65AEx4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A65AEx4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A65AEx4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A65AEx4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A65AEx4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A65AEx4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A65AEx4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A65AEx4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A65AEx4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A65AEx4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A65AEx4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65AEx4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A65AEx4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A65AEx4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A65AEx4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A65AEx4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A65AEx4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A65AEx4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A65AEx4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A65AEx4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A65AEx4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4.cluster0

ARM Cortex-A65AE Cluster CT model.
Type: `Cluster_ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu0.thread0

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu0.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu1.thread0
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu1.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu2.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu2.thread0
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu2.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu3.thread0
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0.cpu3.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A65AEx4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A65AEx4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A65AEx4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A65AEx4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.42 FVP_Base_Cortex-A65AEx4+Cortex-A76AEx4

FVP_Base_Cortex-A65AEx4+Cortex-A76AEx4 contains the following instances:

FVP_Base_Cortex-A65AEx4+Cortex-A76AEx4 instances

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4
Base Platform Compute Subsystem for ARMCortexA65AEx4CT_CortexA76AEx4CT.

Type: `FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl111_clcd_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.refcounter`
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.trusted_watchdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.tzc_400

TrustZone Address Space Controller.
Type: `TZC_400`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.ve_sysregs

Type: `VE_SysRegs`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.virtio_net

VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0

ARM Cortex-A65AE_Cortex-A76AE Cluster CT model.
Type: `Cluster_ARM_Cortex-A65AE_Cortex-A76AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu0.thread0

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu0.thread1

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu1.thread0

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu1.thread1

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu2.thread0

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu2.thread1

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu3.thread0

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu3.thread1

ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu4

ARM Cortex-A76AE CT model.
Type: `ARM_Cortex-A76AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu5

ARM Cortex-A76AE CT model.
Type: `ARM_Cortex-A76AE`.

FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu6

ARM Cortex-A76AE CT model.

Type: `ARM_Cortex-A76AE`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.cpu7`
ARM Cortex-A76AE CT model.
Type: `ARM_Cortex-A76AE`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.subcluster0`
ARM Cortex-A65AE Cluster CT model.
Type: `Subcluster_ARM_Cortex-A65AE`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0.subcluster1`
ARM Cortex-A76AE Cluster CT model.
Type: `Subcluster_ARM_Cortex-A76AE`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.cluster0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.dapmemlogger`
Bus Logger.
Type: `PVBusLogger`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.elfloader`
ELF loader component.
Type: `ElfLoader`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.gic_distributor`
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

`FVP_Base_Cortex_A65AEx4_Cortex_A76AEx4.pctl`
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.43 FVP_Base_Cortex-A65AEx8

FVP_Base_Cortex-A65AEx8 contains the following instances:

FVP_Base_Cortex-A65AEx8 instances

`FVP_Base_Cortex_A65AEx8`
Base Platform Compute Subsystem for ARMCortexA65AEx8CT.
Type: `FVP_Base_Cortex_A65AEx8`.

`FVP_Base_Cortex_A65AEx8.bp`
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A65AEx8.bp.Timer_0_1`
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx8.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx8.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx8.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx8.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx8.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A65AEx8.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx8.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65AEx8.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx8.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65AEx8.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A65AEx8.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A65AEx8.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A65AEx8.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A65AEx8.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A65AEx8.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx8.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx8.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx8.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx8.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65AEx8.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65AEx8.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65AEx8.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A65AEx8.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A65AEx8.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A65AEx8.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx8.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A65AEx8.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A65AEx8.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx8.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65AEx8.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx8.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A65AEx8.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A65AEx8.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A65AEx8.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx8.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A65AEx8.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A65AEx8.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_A65AEx8.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_A65AEx8.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A65AEx8.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65AEx8.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65AEx8.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65AEx8.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A65AEx8.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A65AEx8.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A65AEx8.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A65AEx8.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx8.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A65AEx8.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A65AEx8.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65AEx8.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65AEx8.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65AEx8.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A65AEx8.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65AEx8.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A65AEx8.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65AEx8.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65AEx8.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65AEx8.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65AEx8.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A65AEx8.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A65AEx8.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A65AEx8.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65AEx8.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A65AEx8.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A65AEx8.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A65AEx8.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A65AEx8.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx8.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A65AEx8.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx8.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A65AEx8.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A65AEx8.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A65AEx8.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A65AEx8.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65AEx8.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A65AEx8.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65AEx8.cluster0
ARM Cortex-A65AE Cluster CT model.
Type: `Cluster_ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu0.thread0
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu0.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu1.thread0
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu1.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu2.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu2.thread0
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu2.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu3.thread0
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu3.thread1
ARM Cortex-A65AE CT model.
Type: `ARM_Cortex-A65AE`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu4.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65AEx8.cluster0.cpu4.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu4.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu4.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu4.thread0
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0.cpu4.thread1
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0.cpu5.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A65AEx8.cluster0.cpu5.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu5.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu5.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu5.thread0
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0.cpu5.thread1
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0.cpu6.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A65AEx8.cluster0.cpu6.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu6.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu6.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu6.thread0
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0.cpu6.thread1
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0.cpu7.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A65AEx8.cluster0.cpu7.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu7.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu7.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65AEx8.cluster0.cpu7.thread0
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0.cpu7.thread1
ARM Cortex-A65AE CT model.
Type: ARM_Cortex-A65AE.

FVP_Base_Cortex_A65AEx8.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A65AEx8.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A65AEx8.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A65AEx8.gic_distributor
GIC Interrupt Redistribution Infrastucture component.

Type: GIC_IRI.

FVP_Base_Cortex_A65AEx8.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.44 FVP_Base_Cortex-A65x1

FVP_Base_Cortex-A65x1 contains the following instances:

FVP_Base_Cortex-A65x1 instances

FVP_Base_Cortex_A65x1
Base Platform Compute Subsystem for ARMCortexA65x1CT.
Type: FVP_Base_Cortex_A65x1.

FVP_Base_Cortex_A65x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A65x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A65x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x1.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A65x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A65x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A65x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A65x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A65x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A65x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65x1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65x1.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A65x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A65x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A65x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A65x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A65x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A65x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A65x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A65x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A65x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A65x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A65x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A65x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A65x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A65x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A65x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A65x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A65x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A65x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A65x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A65x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A65x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A65x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A65x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A65x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A65x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A65x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A65x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A65x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A65x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A65x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A65x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A65x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A65x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A65x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A65x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A65x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x1.cluster0

ARM Cortex-A65 Cluster CT model.
Type: `Cluster_ARM_Cortex-A65`.

FVP_Base_Cortex_A65x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x1.cluster0.cpu0.thread0

ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x1.cluster0.cpu0.thread1
ARM Cortex-A65 CT model.
Type: ARM_Cortex-A65.

FVP_Base_Cortex_A65x1.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A65x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A65x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A65x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A65x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.45 FVP_Base_Cortex-A65x2

FVP_Base_Cortex-A65x2 contains the following instances:

FVP_Base_Cortex-A65x2 instances

FVP_Base_Cortex_A65x2
Base Platform Compute Subsystem for ARMCortexA65x2CT.
Type: FVP_Base_Cortex_A65x2.

FVP_Base_Cortex_A65x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A65x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A65x2.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x2.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x2.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A65x2.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x2.bp.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x2.bp.ap_refclk
ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A65x2.bp.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A65x2.bp.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x2.bp.dmc

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A65x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A65x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A65x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A65x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A65x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A65x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A65x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A65x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A65x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A65x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A65x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A65x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A65x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A65x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A65x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A65x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A65x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A65x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A65x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A65x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A65x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A65x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A65x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A65x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A65x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A65x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A65x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A65x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A65x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A65x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A65x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A65x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A65x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A65x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A65x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A65x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A65x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A65x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A65x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A65x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A65x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A65x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A65x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x2.cluster0

ARM Cortex-A65 Cluster CT model.
Type: Cluster_ARM_Cortex-A65.

FVP_Base_Cortex_A65x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A65x2.cluster0.cpu0.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65x2.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65x2.cluster0.cpu0.thread0
ARM Cortex-A65 CT model.
Type: ARM_Cortex-A65.

FVP_Base_Cortex_A65x2.cluster0.cpu0.thread1
ARM Cortex-A65 CT model.
Type: ARM_Cortex-A65.

FVP_Base_Cortex_A65x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A65x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65x2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A65x2.cluster0.cpu1.thread0
ARM Cortex-A65 CT model.
Type: ARM_Cortex-A65.

FVP_Base_Cortex_A65x2.cluster0.cpu1.thread1
ARM Cortex-A65 CT model.
Type: ARM_Cortex-A65.

FVP_Base_Cortex_A65x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A65x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A65x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A65x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.

Type: GIC_IRI.

FVP_Base_Cortex_A65x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.46 FVP_Base_Cortex-A65x4

FVP_Base_Cortex-A65x4 contains the following instances:

FVP_Base_Cortex-A65x4 instances

FVP_Base_Cortex_A65x4
Base Platform Compute Subsystem for ARMCortexA65x4CT.
Type: FVP_Base_Cortex_A65x4.

FVP_Base_Cortex_A65x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A65x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A65x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A65x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A65x4.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A65x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A65x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A65x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A65x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A65x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A65x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A65x4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A65x4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A65x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A65x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A65x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A65x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A65x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A65x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A65x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A65x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A65x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A65x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A65x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A65x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A65x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A65x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A65x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A65x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A65x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A65x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A65x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A65x4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A65x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A65x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A65x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A65x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A65x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A65x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A65x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A65x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A65x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A65x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A65x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A65x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A65x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A65x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A65x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A65x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A65x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A65x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A65x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A65x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A65x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A65x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A65x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A65x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A65x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A65x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A65x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A65x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A65x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A65x4.cluster0

ARM Cortex-A65 Cluster CT model.
Type: `Cluster_ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu0.thread0

ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu0.thread1
ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu1.thread0
ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu1.thread1
ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu2.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu2.thread0
ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu2.thread1
ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A65x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A65x4.cluster0.cpu3.thread0
ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0.cpu3.thread1
ARM Cortex-A65 CT model.
Type: `ARM_Cortex-A65`.

FVP_Base_Cortex_A65x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A65x4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A65x4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A65x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A65x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.47 FVP_Base_Cortex-A710

FVP_Base_Cortex-A710 contains the following instances:

FVP_Base_Cortex-A710 instances

FVP_Base_Cortex_A710
Base Platform Compute Subsystem for ARMCortexA710CT.

Type: `FVP_Base_Cortex_A710`.

`FVP_Base_Cortex_A710.bp`

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A710.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A710.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A710.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A710.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A710.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710.bp.Timer_2_3.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710.bp.Timer_2_3.counter0`

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A710.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A710.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A710.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A710.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A710.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A710.bp.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710.bp.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710.bp.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710.bp.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A710.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A710.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A710.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A710.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A710.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A710.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A710.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A710.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A710.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A710.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A710.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A710.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A710.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A710.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A710.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A710.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A710.bp.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A710.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A710.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A710.bp.pl111_clcd_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_A710.bp.pl180_mci`

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A710.bp.ps2keyboard`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A710.bp.ps2mouse`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A710.bp.psram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710.bp.refcounter`

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A710.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A710.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A710.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A710.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A710.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A710.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A710.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A710.bp.trusted_watchdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A710.bp.tzc_400

TrustZone Address Space Controller.
Type: `TZC_400`.

FVP_Base_Cortex_A710.bp.ve_sysregs

Type: `VE_SysRegs`.

FVP_Base_Cortex_A710.bp.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_Base_Cortex_A710.bp.virtio_net

VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A710.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A710.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A710.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A710.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A710.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A710.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A710.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A710.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A710.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710.cluster0

ARM Cortex-A710 Cluster CT model.
Type: `Cluster_ARM_CortexA710`.

FVP_Base_Cortex_A710.cluster0.cpu0

ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

FVP_Base_Cortex_A710.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A710.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710.cluster0.cpu1

ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

FVP_Base_Cortex_A710.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A710.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710.cluster0.cpu1.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710.cluster0.cpu1.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710.cluster0.cpu2

ARM Cortex-A710 CT model.

Type: `ARM_Cortex-A710`.

`FVP_Base_Cortex_A710.cluster0.cpu2.dtlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Cortex_A710.cluster0.cpu2.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A710.cluster0.cpu2.l1icache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A710.cluster0.cpu2.l2cache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A710.cluster0.cpu3`
ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

`FVP_Base_Cortex_A710.cluster0.cpu3.dtlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Cortex_A710.cluster0.cpu3.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A710.cluster0.cpu3.l1icache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A710.cluster0.cpu3.l2cache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A710.cluster0.cpu4`
ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

`FVP_Base_Cortex_A710.cluster0.cpu4.dtlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Cortex_A710.cluster0.cpu4.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A710.cluster0.cpu4.l1icache`
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710.cluster0.cpu4.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu5
ARM Cortex-A710 CT model.
Type: ARM_Cortex-A710.

FVP_Base_Cortex_A710.cluster0.cpu5.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A710.cluster0.cpu5.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu5.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu5.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu6
ARM Cortex-A710 CT model.
Type: ARM_Cortex-A710.

FVP_Base_Cortex_A710.cluster0.cpu6.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A710.cluster0.cpu6.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu6.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu6.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu7
ARM Cortex-A710 CT model.
Type: ARM_Cortex-A710.

FVP_Base_Cortex_A710.cluster0.cpu7.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A710.cluster0.cpu7.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu7.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0.cpu7.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A710.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A710.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A710.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A710.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.48 FVP_Base_Cortex-A710x1

FVP_Base_Cortex-A710x1 contains the following instances:

FVP_Base_Cortex-A710x1 instances

FVP_Base_Cortex_A710x1
Base Platform Compute Subsystem for ARMCortexA710x1CT.
Type: FVP_Base_Cortex_A710x1.

FVP_Base_Cortex_A710x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A710x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A710x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A710x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A710x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A710x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A710x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A710x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A710x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A710x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A710x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A710x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A710x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A710x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A710x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A710x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A710x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A710x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A710x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A710x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A710x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A710x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x1.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A710x1.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A710x1.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A710x1.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x1.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A710x1.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x1.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_A710x1.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_A710x1.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A710x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A710x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A710x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A710x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A710x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A710x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A710x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A710x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A710x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A710x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A710x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A710x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A710x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x1.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A710x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A710x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A710x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A710x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A710x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A710x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A710x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A710x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A710x1.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A710x1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A710x1.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A710x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A710x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A710x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A710x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A710x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x1.cluster0
ARM Cortex-A710 Cluster CT model.
Type: `Cluster_ARM_CortexA710`.

FVP_Base_Cortex_A710x1.cluster0.cpu0
ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

FVP_Base_Cortex_A710x1.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A710x1.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x1.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x1.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x1.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A710x1.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A710x1.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A710x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A710x1.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.49 FVP_Base_Cortex-A710x2

FVP_Base_Cortex-A710x2 contains the following instances:

FVP_Base_Cortex-A710x2 instances

FVP_Base_Cortex_A710x2
Base Platform Compute Subsystem for ARMCortexA710x2CT.

Type: `FVP_Base_Cortex_A710x2`.

`FVP_Base_Cortex_A710x2.bp`

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A710x2.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A710x2.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x2.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x2.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A710x2.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A710x2.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A710x2.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x2.bp.Timer_2_3.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x2.bp.Timer_2_3.counter0`

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A710x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A710x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A710x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A710x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x2.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x2.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A710x2.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710x2.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710x2.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710x2.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A710x2.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A710x2.bp.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710x2.bp.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710x2.bp.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710x2.bp.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A710x2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A710x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A710x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A710x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A710x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A710x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A710x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A710x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A710x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A710x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A710x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A710x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A710x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A710x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A710x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A710x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A710x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A710x2.bp.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A710x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A710x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A710x2.bp.pl111_clcd_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_A710x2.bp.pl180_mci`

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A710x2.bp.ps2keyboard`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A710x2.bp.ps2mouse`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A710x2.bp.psram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A710x2.bp.refcounter`

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A710x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A710x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A710x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A710x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A710x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710x2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710x2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710x2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710x2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A710x2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A710x2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A710x2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A710x2.bp.trusted_watchdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A710x2.bp.tzc_400

TrustZone Address Space Controller.
Type: `TZC_400`.

FVP_Base_Cortex_A710x2.bp.ve_sysregs

Type: `VE_SysRegs`.

FVP_Base_Cortex_A710x2.bp.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_Base_Cortex_A710x2.bp.virtio_net

VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A710x2.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A710x2.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A710x2.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A710x2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A710x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A710x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A710x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A710x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A710x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x2.cluster0

ARM Cortex-A710 Cluster CT model.
Type: `Cluster_ARM_CortexA710`.

FVP_Base_Cortex_A710x2.cluster0.cpu0

ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

FVP_Base_Cortex_A710x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A710x2.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x2.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x2.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x2.cluster0.cpu1

ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

FVP_Base_Cortex_A710x2.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A710x2.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x2.cluster0.cpu1.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x2.cluster0.cpu1.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x2.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A710x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A710x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A710x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A710x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.50 FVP_Base_Cortex-A710x4

FVP_Base_Cortex-A710x4 contains the following instances:

FVP_Base_Cortex-A710x4 instances

FVP_Base_Cortex_A710x4
Base Platform Compute Subsystem for ARMCortexA710x4CT.
Type: FVP_Base_Cortex_A710x4.

FVP_Base_Cortex_A710x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A710x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A710x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A710x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A710x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A710x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A710x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A710x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A710x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A710x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x4.bp.dram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A710x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A710x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A710x4.bp.exclusive_monitor
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A710x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A710x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A710x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A710x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A710x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_Base_Cortex_A710x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Cortex_A710x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A710x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A710x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A710x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A710x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A710x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A710x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A710x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A710x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A710x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A710x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A710x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A710x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A710x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A710x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A710x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A710x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A710x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A710x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A710x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A710x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A710x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A710x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A710x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A710x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A710x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A710x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A710x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A710x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A710x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A710x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A710x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A710x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A710x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A710x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A710x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A710x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A710x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A710x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A710x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A710x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A710x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A710x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A710x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A710x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A710x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A710x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A710x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A710x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A710x4.cluster0

ARM Cortex-A710 Cluster CT model.
Type: Cluster_ARM_CortexA710.

FVP_Base_Cortex_A710x4.cluster0.cpu0

ARM Cortex-A710 CT model.
Type: ARM_Cortex-A710.

FVP_Base_Cortex_A710x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A710x4.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu1
ARM Cortex-A710 CT model.
Type: ARM_Cortex-A710.

FVP_Base_Cortex_A710x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A710x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu2
ARM Cortex-A710 CT model.
Type: ARM_Cortex-A710.

FVP_Base_Cortex_A710x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A710x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A710x4.cluster0.cpu3
ARM Cortex-A710 CT model.
Type: `ARM_Cortex-A710`.

FVP_Base_Cortex_A710x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A710x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x4.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A710x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A710x4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A710x4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A710x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A710x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.51 FVP_Base_Cortex-A72x1

FVP_Base_Cortex-A72x1 contains the following instances:

FVP_Base_Cortex-A72x1 instances

FVP_Base_Cortex_A72x1
Base Platform Compute Subsystem for ARMCortexA72x1CT.
Type: `FVP_Base_Cortex_A72x1`.

FVP_Base_Cortex_A72x1.bp
Peripherals and address map for the Base Platform.

Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A72x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A72x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A72x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A72x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A72x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A72x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1.bp.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A72x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A72x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A72x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A72x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A72x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A72x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A72x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A72x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A72x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A72x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A72x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A72x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A72x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A72x1.bp.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A72x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A72x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A72x1.bp.pl111_clcd_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_A72x1.bp.pl180_mci`

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A72x1.bp.ps2keyboard`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A72x1.bp.ps2mouse`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A72x1.bp.psram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A72x1.bp.refcounter`

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

`FVP_Base_Cortex_A72x1.bp.reset_or`

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A72x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A72x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A72x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A72x1.bp.sram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A72x1.bp.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A72x1.bp.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A72x1.bp.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A72x1.bp.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A72x1.bp.trusted_key_storage`
Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

`FVP_Base_Cortex_A72x1.bp.trusted_nv_counter`
Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

`FVP_Base_Cortex_A72x1.bp.trusted_rng`
Random Number Generator unit.
Type: `RandomNumberGenerator`.

`FVP_Base_Cortex_A72x1.bp.trusted_watchdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_Base_Cortex_A72x1.bp.tzc_400`
TrustZone Address Space Controller.
Type: `TZC_400`.

`FVP_Base_Cortex_A72x1.bp.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_Base_Cortex_A72x1.bp.vedcc`
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

`FVP_Base_Cortex_A72x1.bp.virtio_net`
VirtioNet device over MMIO transport.
Type: `VirtioNetMMIO`.

FVP_Base_Cortex_A72x1.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1.bp.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_Base_Cortex_A72x1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1.bp.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_Base_Cortex_A72x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1.bp.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_Base_Cortex_A72x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_Base_Cortex_A72x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A72x1.bp.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A72x1.cci400

Cache Coherent Interconnect for AXI4 ACE.

Type: CCI400.

FVP_Base_Cortex_A72x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A72x1.cluster0

ARM Cortex-A72 Cluster CT model.
Type: Cluster_ARM_Cortex-A72.

FVP_Base_Cortex_A72x1.cluster0.cpu0

ARM Cortex-A72 CT model.
Type: ARM_Cortex-A72.

FVP_Base_Cortex_A72x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A72x1.cluster0.cpu0.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x1.cluster0.cpu0.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x1.cluster0.l2_cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x1.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A72x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A72x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A72x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.52 FVP_Base_Cortex-A72x1-A53x1

FVP_Base_Cortex-A72x1-A53x1 contains the following instances:

FVP_Base_Cortex-A72x1-A53x1 instances

FVP_Base_Cortex_A72x1_A53x1
Base Platform Compute Subsystem for ARMCortexA72x1CT and ARMCortexA53x1CT.
Type: `FVP_Base_Cortex_A72x1_A53x1`.

FVP_Base_Cortex_A72x1_A53x1.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x1_A53x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x1_A53x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A72x1_A53x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A72x1_A53x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A72x1_A53x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A72x1_A53x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x1_A53x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x1_A53x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A72x1_A53x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x1_A53x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x1_A53x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x1_A53x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x1_A53x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A72x1_A53x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A72x1_A53x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x1_A53x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A72x1_A53x1.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A72x1_A53x1.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A72x1_A53x1.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A72x1_A53x1.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A72x1_A53x1.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A72x1_A53x1.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A72x1_A53x1.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x1_A53x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1_A53x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A72x1_A53x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A72x1_A53x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x1_A53x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1_A53x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x1_A53x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1_A53x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A72x1_A53x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A72x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A72x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x1_A53x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1_A53x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A72x1_A53x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A72x1_A53x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A72x1_A53x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A72x1_A53x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A72x1_A53x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A72x1_A53x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A72x1_A53x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A72x1_A53x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A72x1_A53x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A72x1_A53x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x1_A53x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x1_A53x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x1_A53x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x1_A53x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A72x1_A53x1.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A72x1_A53x1.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A72x1_A53x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x1_A53x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A72x1_A53x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A72x1_A53x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A72x1_A53x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A72x1_A53x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A72x1_A53x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A72x1_A53x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A72x1_A53x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A72x1_A53x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A72x1_A53x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A72x1_A53x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A72x1_A53x1.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x1_A53x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x1_A53x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A72x1_A53x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x1_A53x1.cluster0

ARM Cortex-A72 Cluster CT model.
Type: `Cluster_ARM_Cortex-A72`.

FVP_Base_Cortex_A72x1_A53x1.cluster0.cpu0

ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x1_A53x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x1_A53x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x1_A53x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x1_A53x1.cluster0.l2_cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x1_A53x1.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1_A53x1.cluster1

ARM Cortex-A53 Cluster CT model.

Type: Cluster_ARM_Cortex-A53.

FVP_Base_Cortex_A72x1_A53x1.cluster1.cpu0

ARM Cortex-A53 CT model.

Type: ARM_Cortex-A53.

FVP_Base_Cortex_A72x1_A53x1.cluster1.cpu0.dtlb

TLB - instruction, data or unified.

Type: TlbCadi.

FVP_Base_Cortex_A72x1_A53x1.cluster1.cpu0.l1dcache

PV Cache.

Type: PVCache.

FVP_Base_Cortex_A72x1_A53x1.cluster1.cpu0.l1icache

PV Cache.

Type: PVCache.

FVP_Base_Cortex_A72x1_A53x1.cluster1.l2_cache

PV Cache.

Type: PVCache.

FVP_Base_Cortex_A72x1_A53x1.cluster1_labeller

Type: Labeller.

FVP_Base_Cortex_A72x1_A53x1.dapmemlogger

Bus Logger.

Type: PVBusLogger.

FVP_Base_Cortex_A72x1_A53x1.elfloader

ELF loader component.

Type: ElfLoader.

FVP_Base_Cortex_A72x1_A53x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.

Type: GIC_IRI.

FVP_Base_Cortex_A72x1_A53x1.pctl

Base Platforms Power Controller.

Type: Base_PowerController.

7.53 FVP_Base_Cortex-A72x2

FVP_Base_Cortex-A72x2 contains the following instances:

FVP_Base_Cortex-A72x2 instances

FVP_Base_Cortex_A72x2

Base Platform Compute Subsystem for ARMCortexA72x2CT.
Type: `FVP_Base_Cortex_A72x2`.

FVP_Base_Cortex_A72x2.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A72x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A72x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A72x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A72x2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A72x2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A72x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A72x2.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A72x2.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x2.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A72x2.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A72x2.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A72x2.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A72x2.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A72x2.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A72x2.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A72x2.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A72x2.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A72x2.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A72x2.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A72x2.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A72x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A72x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A72x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A72x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A72x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A72x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A72x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A72x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A72x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A72x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A72x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A72x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A72x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A72x2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A72x2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A72x2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A72x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A72x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A72x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A72x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A72x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A72x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A72x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A72x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A72x2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A72x2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A72x2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A72x2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A72x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2.cluster0

ARM Cortex-A72 Cluster CT model.
Type: `Cluster_ARM_Cortex-A72`.

FVP_Base_Cortex_A72x2.cluster0.cpu0

ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x2.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2.cluster0.cpu1

ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x2.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x2.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x2.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A72x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A72x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A72x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A72x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.54 FVP_Base_Cortex-A72x2-A53x4

FVP_Base_Cortex-A72x2-A53x4 contains the following instances:

FVP_Base_Cortex-A72x2-A53x4 instances

FVP_Base_Cortex_A72x2_A53x4
Base Platform Compute Subsystem for ARMCortexA72x2CT and ARMCortexA53x4CT.
Type: FVP_Base_Cortex_A72x2_A53x4.

FVP_Base_Cortex_A72x2_A53x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2_A53x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x2_A53x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A72x2_A53x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A72x2_A53x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A72x2_A53x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A72x2_A53x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A72x2_A53x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x2_A53x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x2_A53x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x2_A53x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x2_A53x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A72x2_A53x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A72x2_A53x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x2_A53x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A72x2_A53x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x2_A53x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x2_A53x4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A72x2_A53x4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A72x2_A53x4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A72x2_A53x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x2_A53x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x2_A53x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A72x2_A53x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A72x2_A53x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x2_A53x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x2_A53x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A72x2_A53x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A72x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Cortex_A72x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x2_A53x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A72x2_A53x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A72x2_A53x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A72x2_A53x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A72x2_A53x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A72x2_A53x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A72x2_A53x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x2_A53x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x2_A53x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A72x2_A53x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x2_A53x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A72x2_A53x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x2_A53x4.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x2_A53x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x2_A53x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x2_A53x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A72x2_A53x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A72x2_A53x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A72x2_A53x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x2_A53x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A72x2_A53x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A72x2_A53x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A72x2_A53x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A72x2_A53x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A72x2_A53x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A72x2_A53x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A72x2_A53x4.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A72x2_A53x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A72x2_A53x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A72x2_A53x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A72x2_A53x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x2_A53x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A72x2_A53x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x2_A53x4.cluster0
ARM Cortex-A72 Cluster CT model.
Type: `Cluster_ARM_Cortex-A72`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu0
ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu1
ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2_A53x4.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2_A53x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A72x2_A53x4.cluster1
ARM Cortex-A53 Cluster CT model.
Type: `Cluster_ARM_Cortex-A53`.

FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu0
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu0.dtlb
TLB - instruction, data or unified.

Type: `TlbCadi`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu0.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu0.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu1`

ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu1.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu1.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu1.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu2`

ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu2.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu2.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu2.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu3`

ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu3.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu3.l1dcache`

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x2_A53x4.cluster1.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x2_A53x4.cluster1.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x2_A53x4.cluster1_labeller
Type: Labeller.

FVP_Base_Cortex_A72x2_A53x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A72x2_A53x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A72x2_A53x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A72x2_A53x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.55 FVP_Base_Cortex-A72x4

FVP_Base_Cortex-A72x4 contains the following instances:

FVP_Base_Cortex-A72x4 instances

FVP_Base_Cortex_A72x4
Base Platform Compute Subsystem for ARMCortexA72x4CT.
Type: FVP_Base_Cortex_A72x4.

FVP_Base_Cortex_A72x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A72x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A72x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A72x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A72x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A72x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A72x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A72x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A72x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A72x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A72x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A72x4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A72x4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A72x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A72x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A72x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A72x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A72x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A72x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A72x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A72x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A72x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A72x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A72x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Cortex_A72x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A72x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A72x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A72x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A72x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A72x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A72x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A72x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A72x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A72x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A72x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A72x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A72x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A72x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A72x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A72x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A72x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A72x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A72x4.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A72x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A72x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A72x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A72x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A72x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4.cluster0

ARM Cortex-A72 Cluster CT model.
Type: Cluster_ARM_Cortex-A72.

FVP_Base_Cortex_A72x4.cluster0.cpu0

ARM Cortex-A72 CT model.
Type: ARM_Cortex-A72.

FVP_Base_Cortex_A72x4.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4.cluster0.cpu1
ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4.cluster0.cpu2
ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4.cluster0.cpu3
ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A72x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A72x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A72x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A72x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.56 FVP_Base_Cortex-A72x4-A53x4

FVP_Base_Cortex-A72x4-A53x4 contains the following instances:

FVP_Base_Cortex-A72x4-A53x4 instances

FVP_Base_Cortex_A72x4_A53x4
Base Platform Compute Subsystem for ARMCortexA72x4CT and ARMCortexA53x4CT.
Type: FVP_Base_Cortex_A72x4_A53x4.

FVP_Base_Cortex_A72x4_A53x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4_A53x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A72x4_A53x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A72x4_A53x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A72x4_A53x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.bp.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A72x4_A53x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A72x4_A53x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A72x4_A53x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x4_A53x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x4_A53x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x4_A53x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x4_A53x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A72x4_A53x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A72x4_A53x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A72x4_A53x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A72x4_A53x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A72x4_A53x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A72x4_A53x4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A72x4_A53x4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A72x4_A53x4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A72x4_A53x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A72x4_A53x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A72x4_A53x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_A72x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A72x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A72x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A72x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A72x4_A53x4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A72x4_A53x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A72x4_A53x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A72x4_A53x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A72x4_A53x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x4_A53x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A72x4_A53x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A72x4_A53x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x4_A53x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A72x4_A53x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A72x4_A53x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A72x4_A53x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A72x4_A53x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A72x4_A53x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A72x4_A53x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4_A53x4.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4_A53x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4_A53x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A72x4_A53x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A72x4_A53x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A72x4_A53x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A72x4_A53x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A72x4_A53x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A72x4_A53x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A72x4_A53x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A72x4_A53x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A72x4_A53x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A72x4_A53x4.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A72x4_A53x4.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A72x4_A53x4.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A72x4_A53x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A72x4_A53x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A72x4_A53x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A72x4_A53x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A72x4_A53x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A72x4_A53x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A72x4_A53x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A72x4_A53x4.cluster0

ARM Cortex-A72 Cluster CT model.
Type: `Cluster_ARM_Cortex-A72`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu0

ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu1

ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu1.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu2

ARM Cortex-A72 CT model.
Type: `ARM_Cortex-A72`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu2.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu2.l1dcache

PV Cache.

Type: PVCache.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu3

ARM Cortex-A72 CT model.
Type: ARM_Cortex-A72.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu3.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4_A53x4.cluster0.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4_A53x4.cluster0.l2_cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4_A53x4.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A72x4_A53x4.cluster1

ARM Cortex-A53 Cluster CT model.
Type: Cluster_ARM_Cortex-A53.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu0

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu0.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu0.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu1

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu2
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu3
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A72x4_A53x4.cluster1_labeller
Type: `Labeller`.

FVP_Base_Cortex_A72x4_A53x4.dapmemlogger
Bus Logger.

Type: PVBusLogger.

FVP_Base_Cortex_A72x4_A53x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A72x4_A53x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A72x4_A53x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.57 FVP_Base_Cortex-A73x1

FVP_Base_Cortex-A73x1 contains the following instances:

FVP_Base_Cortex-A73x1 instances

FVP_Base_Cortex_A73x1

Base Platform Compute Subsystem for ARMCortexA73x1CT.
Type: FVP_Base_Cortex_A73x1.

FVP_Base_Cortex_A73x1.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A73x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A73x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A73x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A73x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A73x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A73x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A73x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A73x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A73x1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A73x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A73x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A73x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A73x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A73x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A73x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A73x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A73x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A73x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A73x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A73x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A73x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A73x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A73x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A73x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A73x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A73x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A73x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A73x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A73x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A73x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A73x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A73x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A73x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A73x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A73x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A73x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A73x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A73x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A73x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A73x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A73x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A73x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A73x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A73x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A73x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A73x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A73x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A73x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A73x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A73x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A73x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A73x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1.cluster0

ARM Cortex-A73 Cluster CT model.
Type: Cluster_ARM_Cortex-A73.

FVP_Base_Cortex_A73x1.cluster0.cpu0

ARM Cortex-A73 CT model.
Type: ARM_Cortex-A73.

FVP_Base_Cortex_A73x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A73x1.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x1.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x1.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x1.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A73x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A73x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A73x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A73x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.58 FVP_Base_Cortex-A73x1-A53x1

FVP_Base_Cortex-A73x1-A53x1 contains the following instances:

FVP_Base_Cortex-A73x1-A53x1 instances

FVP_Base_Cortex_A73x1_A53x1
Base Platform Compute Subsystem for ARMCortexA73x1CT and ARMCortexA53x1CT.
Type: FVP_Base_Cortex_A73x1_A53x1.

FVP_Base_Cortex_A73x1_A53x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x1_A53x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x1_A53x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A73x1_A53x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A73x1_A53x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A73x1_A53x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A73x1_A53x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A73x1_A53x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x1_A53x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x1_A53x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x1_A53x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x1_A53x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A73x1_A53x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A73x1_A53x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A73x1_A53x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A73x1_A53x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A73x1_A53x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A73x1_A53x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A73x1_A53x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A73x1_A53x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A73x1_A53x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x1_A53x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x1_A53x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_A73x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A73x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A73x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A73x1_A53x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A73x1_A53x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A73x1_A53x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A73x1_A53x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A73x1_A53x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A73x1_A53x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x1_A53x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A73x1_A53x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A73x1_A53x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x1_A53x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x1_A53x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x1_A53x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A73x1_A53x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x1_A53x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A73x1_A53x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1_A53x1.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1_A53x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1_A53x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x1_A53x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A73x1_A53x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A73x1_A53x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A73x1_A53x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x1_A53x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A73x1_A53x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A73x1_A53x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A73x1_A53x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A73x1_A53x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A73x1_A53x1.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A73x1_A53x1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A73x1_A53x1.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A73x1_A53x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A73x1_A53x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A73x1_A53x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A73x1_A53x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x1_A53x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A73x1_A53x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x1_A53x1.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x1_A53x1.cluster0

ARM Cortex-A73 Cluster CT model.
Type: `Cluster_ARM_Cortex-A73`.

FVP_Base_Cortex_A73x1_A53x1.cluster0.cpu0

ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x1_A53x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x1_A53x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x1_A53x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x1_A53x1.cluster0.l2_cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x1_A53x1.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A73x1_A53x1.cluster1

ARM Cortex-A53 Cluster CT model.
Type: `Cluster_ARM_Cortex-A53`.

FVP_Base_Cortex_A73x1_A53x1.cluster1.cpu0

ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A73x1_A53x1.cluster1.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x1_A53x1.cluster1.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x1_A53x1.cluster1.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x1_A53x1.cluster1.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x1_A53x1.cluster1_labeller
Type: Labeller.

FVP_Base_Cortex_A73x1_A53x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A73x1_A53x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A73x1_A53x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A73x1_A53x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.59 FVP_Base_Cortex-A73x2

FVP_Base_Cortex-A73x2 contains the following instances:

FVP_Base_Cortex-A73x2 instances

FVP_Base_Cortex_A73x2
Base Platform Compute Subsystem for ARMCortexA73x2CT.
Type: FVP_Base_Cortex_A73x2.

FVP_Base_Cortex_A73x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A73x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A73x2.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x2.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x2.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A73x2.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x2.bp.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x2.bp.ap_refclk
ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A73x2.bp.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A73x2.bp.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2.bp.dmc

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A73x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A73x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A73x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A73x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A73x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A73x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A73x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A73x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A73x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A73x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A73x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A73x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A73x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A73x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A73x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A73x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A73x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A73x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A73x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A73x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A73x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A73x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A73x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A73x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A73x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A73x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A73x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A73x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A73x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A73x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A73x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A73x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A73x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A73x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A73x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A73x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A73x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A73x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A73x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A73x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A73x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A73x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A73x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A73x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A73x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2.cluster0

ARM Cortex-A73 Cluster CT model.
Type: Cluster_ARM_Cortex-A73.

FVP_Base_Cortex_A73x2.cluster0.cpu0

ARM Cortex-A73 CT model.
Type: ARM_Cortex-A73.

FVP_Base_Cortex_A73x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A73x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2.cluster0.cpu1
ARM Cortex-A73 CT model.
Type: ARM_Cortex-A73.

FVP_Base_Cortex_A73x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A73x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A73x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A73x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A73x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A73x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.60 FVP_Base_Cortex-A73x2-A53x4

FVP_Base_Cortex-A73x2-A53x4 contains the following instances:

FVP_Base_Cortex-A73x2-A53x4 instances

FVP_Base_Cortex_A73x2_A53x4
Base Platform Compute Subsystem for ARMCortexA73x2CT and ARMCortexA53x4CT.
Type: `FVP_Base_Cortex_A73x2_A53x4`.

FVP_Base_Cortex_A73x2_A53x4.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x2_A53x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x2_A53x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A73x2_A53x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A73x2_A53x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A73x2_A53x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A73x2_A53x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2_A53x4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x2_A53x4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A73x2_A53x4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x2_A53x4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x2_A53x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x2_A53x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x2_A53x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A73x2_A53x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A73x2_A53x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A73x2_A53x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A73x2_A53x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A73x2_A53x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A73x2_A53x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A73x2_A53x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A73x2_A53x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A73x2_A53x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A73x2_A53x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A73x2_A53x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2_A53x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A73x2_A53x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A73x2_A53x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A73x2_A53x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2_A53x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A73x2_A53x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2_A53x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A73x2_A53x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A73x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A73x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A73x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A73x2_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A73x2_A53x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A73x2_A53x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A73x2_A53x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A73x2_A53x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A73x2_A53x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A73x2_A53x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A73x2_A53x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A73x2_A53x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A73x2_A53x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A73x2_A53x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A73x2_A53x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A73x2_A53x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x2_A53x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x2_A53x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x2_A53x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x2_A53x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A73x2_A53x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A73x2_A53x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A73x2_A53x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x2_A53x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A73x2_A53x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A73x2_A53x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A73x2_A53x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A73x2_A53x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A73x2_A53x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A73x2_A53x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A73x2_A53x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A73x2_A53x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A73x2_A53x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A73x2_A53x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A73x2_A53x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x2_A53x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x2_A53x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A73x2_A53x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x2_A53x4.cluster0

ARM Cortex-A73 Cluster CT model.
Type: `Cluster_ARM_Cortex-A73`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu0

ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu1

ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A73x2_A53x4.cluster1
ARM Cortex-A53 Cluster CT model.
Type: `Cluster_ARM_Cortex-A53`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu0
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu1
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu1.l1icache
PV Cache.

Type: PVCache.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu2

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu2.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu2.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu3

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu3.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2_A53x4.cluster1.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2_A53x4.cluster1.l2_cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x2_A53x4.cluster1_labeller

Type: Labeller.

FVP_Base_Cortex_A73x2_A53x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A73x2_A53x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A73x2_A53x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A73x2_A53x4.pctl

Base Platforms Power Controller.
Type: `Base_PowerController`.

7.61 FVP_Base_Cortex-A73x4

FVP_Base_Cortex-A73x4 contains the following instances:

FVP_Base_Cortex-A73x4 instances

FVP_Base_Cortex_A73x4

Base Platform Compute Subsystem for ARMCortexA73x4CT.
Type: `FVP_Base_Cortex_A73x4`.

FVP_Base_Cortex_A73x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A73x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A73x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A73x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A73x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A73x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A73x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A73x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A73x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A73x4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A73x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A73x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A73x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A73x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A73x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A73x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A73x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A73x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A73x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A73x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A73x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A73x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A73x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A73x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A73x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A73x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A73x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A73x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A73x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A73x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A73x4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A73x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A73x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A73x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A73x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A73x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A73x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A73x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A73x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A73x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A73x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A73x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A73x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A73x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A73x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A73x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A73x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A73x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A73x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A73x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A73x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A73x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A73x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A73x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A73x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A73x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A73x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A73x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4.cluster0

ARM Cortex-A73 Cluster CT model.
Type: `Cluster_ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4.cluster0.cpu0

ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.cpu1

ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.cpu2
ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.cpu3
ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A73x4.dapmemlogger
Bus Logger.

Type: PVBusLogger.

FVP_Base_Cortex_A73x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A73x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A73x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.62 FVP_Base_Cortex-A73x4-A53x4

FVP_Base_Cortex-A73x4-A53x4 contains the following instances:

FVP_Base_Cortex-A73x4-A53x4 instances

FVP_Base_Cortex_A73x4_A53x4

Base Platform Compute Subsystem for ARMCortexA73x4CT and ARMCortexA53x4CT.
Type: FVP_Base_Cortex_A73x4_A53x4.

FVP_Base_Cortex_A73x4_A53x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x4_A53x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A73x4_A53x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A73x4_A53x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A73x4_A53x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4_A53x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4_A53x4.bp.dram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4_A53x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A73x4_A53x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A73x4_A53x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4_A53x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4_A53x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4_A53x4.bp.exclusive_monitor
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A73x4_A53x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A73x4_A53x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A73x4_A53x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A73x4_A53x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A73x4_A53x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_Base_Cortex_A73x4_A53x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Cortex_A73x4_A53x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A73x4_A53x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A73x4_A53x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A73x4_A53x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A73x4_A53x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A73x4_A53x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A73x4_A53x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A73x4_A53x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A73x4_A53x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A73x4_A53x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A73x4_A53x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A73x4_A53x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A73x4_A53x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A73x4_A53x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A73x4_A53x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x4_A53x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A73x4_A53x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A73x4_A53x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x4_A53x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x4_A53x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A73x4_A53x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A73x4_A53x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A73x4_A53x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x4_A53x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A73x4_A53x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A73x4_A53x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A73x4_A53x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x4_A53x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x4_A53x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x4_A53x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A73x4_A53x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A73x4_A53x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A73x4_A53x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A73x4_A53x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A73x4_A53x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A73x4_A53x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A73x4_A53x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A73x4_A53x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A73x4_A53x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A73x4_A53x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A73x4_A53x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A73x4_A53x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A73x4_A53x4.bp.virtiop9device_labeller

Type: `Labeller`.

FVP_Base_Cortex_A73x4_A53x4.bp.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_Base_Cortex_A73x4_A53x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_Base_Cortex_A73x4_A53x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A73x4_A53x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A73x4_A53x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A73x4_A53x4.cluster0

ARM Cortex-A73 Cluster CT model.
Type: `Cluster_ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu0
ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu1
ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu2
ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu3
ARM Cortex-A73 CT model.
Type: `ARM_Cortex-A73`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A73x4_A53x4.cluster1
ARM Cortex-A53 Cluster CT model.
Type: `Cluster_ARM_Cortex-A53`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu0
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu1
ARM Cortex-A53 CT model.
Type: `ARM_Cortex-A53`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu1.l1icache
PV Cache.

Type: PVCache.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu2

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu2.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu2.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu3

ARM Cortex-A53 CT model.
Type: ARM_Cortex-A53.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu3.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x4_A53x4.cluster1.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x4_A53x4.cluster1.l2_cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A73x4_A53x4.cluster1_labeller

Type: Labeller.

FVP_Base_Cortex_A73x4_A53x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A73x4_A53x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A73x4_A53x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A73x4_A53x4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.63 FVP_Base_Cortex-A75

FVP_Base_Cortex-A75 contains the following instances:

FVP_Base_Cortex-A75 instances

FVP_Base_Cortex_A75
Base Platform Compute Subsystem for ARMCortexA75CT.
Type: `FVP_Base_Cortex_A75`.

FVP_Base_Cortex_A75.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A75.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A75.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A75.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A75.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A75.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A75.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A75.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A75.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A75.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A75.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A75.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A75.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A75.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A75.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A75.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A75.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A75.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A75.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A75.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A75.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A75.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A75.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A75.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A75.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A75.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A75.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A75.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A75.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A75.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A75.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A75.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A75.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A75.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A75.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A75.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A75.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A75.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A75.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A75.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A75.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A75.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A75.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A75.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A75.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A75.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A75.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A75.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A75.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A75.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A75.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A75.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A75.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A75.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A75.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A75.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A75.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75.cluster0

ARM Cortex-A75 Cluster CT model.
Type: `Cluster_ARM_Cortex-A75`.

FVP_Base_Cortex_A75.cluster0.cpu0

ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A75.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75.cluster0.cpu1
ARM Cortex-A75 CT model.
Type: ARM_Cortex-A75.

FVP_Base_Cortex_A75.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A75.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu2
ARM Cortex-A75 CT model.
Type: ARM_Cortex-A75.

FVP_Base_Cortex_A75.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A75.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu3
ARM Cortex-A75 CT model.
Type: ARM_Cortex-A75.

FVP_Base_Cortex_A75.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A75.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A75.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A75.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A75.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A75.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.64 FVP_Base_Cortex-A75x1

FVP_Base_Cortex-A75x1 contains the following instances:

FVP_Base_Cortex-A75x1 instances

FVP_Base_Cortex_A75x1
Base Platform Compute Subsystem for ARMCortexA75x1CT.
Type: FVP_Base_Cortex_A75x1.

FVP_Base_Cortex_A75x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A75x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A75x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A75x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A75x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A75x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A75x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A75x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A75x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A75x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A75x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A75x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A75x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A75x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A75x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A75x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A75x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A75x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A75x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A75x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A75x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A75x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A75x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A75x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A75x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A75x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A75x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A75x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A75x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A75x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A75x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A75x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A75x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A75x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A75x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x1.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A75x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A75x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A75x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A75x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A75x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A75x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A75x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A75x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A75x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A75x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A75x1.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A75x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A75x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A75x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A75x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A75x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x1.cluster0

ARM Cortex-A75 Cluster CT model.
Type: Cluster_ARM_Cortex-A75.

FVP_Base_Cortex_A75x1.cluster0.cpu0

ARM Cortex-A75 CT model.
Type: ARM_Cortex-A75.

FVP_Base_Cortex_A75x1.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75x1.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x1.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x1.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x1.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A75x1.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A75x1.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A75x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A75x1.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.65 FVP_Base_Cortex-A75x2

FVP_Base_Cortex-A75x2 contains the following instances:

FVP_Base_Cortex-A75x2 instances

FVP_Base_Cortex_A75x2
Base Platform Compute Subsystem for ARMCortexA75x2CT.
Type: `FVP_Base_Cortex_A75x2`.

FVP_Base_Cortex_A75x2.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A75x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).

Type: SP804_Timer.

FVP_Base_Cortex_A75x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A75x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x2.bp.ap_refclk

ARM Generic Timer.

Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A75x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A75x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A75x2.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A75x2.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A75x2.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A75x2.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A75x2.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A75x2.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A75x2.bp.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A75x2.bp.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A75x2.bp.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A75x2.bp.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_Base_Cortex_A75x2.bp.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A75x2.bp.flash1`
Intel Strata Flash J3 LISA+ model.

Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A75x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A75x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A75x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A75x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A75x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A75x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A75x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A75x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A75x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A75x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A75x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A75x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A75x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A75x2.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A75x2.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A75x2.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A75x2.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A75x2.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A75x2.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A75x2.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A75x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A75x2.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A75x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A75x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A75x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A75x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A75x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A75x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A75x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A75x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A75x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A75x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x2.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x2.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A75x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A75x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A75x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A75x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A75x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A75x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A75x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A75x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A75x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A75x2.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A75x2.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A75x2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A75x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A75x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A75x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A75x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A75x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x2.cluster0
ARM Cortex-A75 Cluster CT model.
Type: `Cluster_ARM_Cortex-A75`.

FVP_Base_Cortex_A75x2.cluster0.cpu0
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A75x2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75x2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x2.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x2.cluster0.cpu1
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A75x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75x2.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x2.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x2.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A75x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A75x2.elfloader
ELF loader component.

Type: ElfLoader.

FVP_Base_Cortex_A75x2.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A75x2.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.66 FVP_Base_Cortex-A75x4

FVP_Base_Cortex-A75x4 contains the following instances:

FVP_Base_Cortex-A75x4 instances

FVP_Base_Cortex_A75x4

Base Platform Compute Subsystem for ARMCortexA75x4CT.
Type: FVP_Base_Cortex_A75x4.

FVP_Base_Cortex_A75x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A75x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A75x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A75x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A75x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A75x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A75x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x4.bp.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A75x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A75x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x4.bp.exclusive_monitor
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A75x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A75x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A75x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A75x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A75x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_Base_Cortex_A75x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Cortex_A75x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A75x4.bp.hdlcd0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A75x4.bp.hdlcd0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A75x4.bp.hdlcd0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A75x4.bp.hdlcd0_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_A75x4.bp.hostbridge`

Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A75x4.bp.mmc`

Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A75x4.bp.nontrustedrom`

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A75x4.bp.nontrustedromloader`

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A75x4.bp.pl011_uart0`

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A75x4.bp.pl011_uart0.clk_divider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A75x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A75x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A75x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A75x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A75x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A75x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A75x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A75x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A75x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A75x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A75x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).

Type: PL180_MCI.

FVP_Base_Cortex_A75x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A75x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A75x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A75x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A75x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A75x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A75x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A75x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A75x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A75x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A75x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A75x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A75x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A75x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A75x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A75x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A75x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A75x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A75x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A75x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A75x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A75x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A75x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A75x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A75x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A75x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A75x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A75x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A75x4.cluster0

ARM Cortex-A75 Cluster CT model.
Type: `Cluster_ARM_Cortex-A75`.

FVP_Base_Cortex_A75x4.cluster0.cpu0

ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A75x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu1
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A75x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu2
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A75x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu2.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu3
ARM Cortex-A75 CT model.
Type: `ARM_Cortex-A75`.

FVP_Base_Cortex_A75x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A75x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A75x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A75x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A75x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A75x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A75x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A75x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.67 FVP_Base_Cortex-A76

FVP_Base_Cortex-A76 contains the following instances:

FVP_Base_Cortex-A76 instances

FVP_Base_Cortex_A76
Base Platform Compute Subsystem for ARMCortexA76CT.
Type: FVP_Base_Cortex_A76.

FVP_Base_Cortex_A76.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A76.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A76.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A76.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A76.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A76.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A76.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A76.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A76.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A76.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A76.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A76.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A76.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A76.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A76.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A76.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A76.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A76.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A76.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A76.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A76.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A76.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A76.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A76.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A76.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A76.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A76.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A76.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A76.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A76.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A76.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A76.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A76.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A76.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A76.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A76.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A76.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A76.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A76.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A76.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A76.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A76.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A76.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A76.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A76.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A76.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A76.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A76.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76.cluster0

ARM Cortex-A76 Cluster CT model.
Type: Cluster_ARM_Cortex-A76.

FVP_Base_Cortex_A76.cluster0.cpu0

ARM Cortex-A76 CT model.
Type: ARM_Cortex-A76.

FVP_Base_Cortex_A76.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu1
ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A76.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu2
ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A76.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76.cluster0.cpu3
ARM Cortex-A76 CT model.
Type: ARM_Cortex-A76.

FVP_Base_Cortex_A76.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A76.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A76.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A76.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A76.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A76.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.68 FVP_Base_Cortex-A76AEx2

FVP_Base_Cortex-A76AEx2 contains the following instances:

FVP_Base_Cortex-A76AEx2 instances

FVP_Base_Cortex_A76AEx2
Base Platform Compute Subsystem for ARMCortexA76AEx2CT.

Type: `FVP_Base_Cortex_A76AEx2`.

`FVP_Base_Cortex_A76AEx2.bp`

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_2_3.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76AEx2.bp.Timer_2_3.counter0`

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A76AEx2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A76AEx2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A76AEx2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A76AEx2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76AEx2.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76AEx2.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76AEx2.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76AEx2.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76AEx2.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76AEx2.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A76AEx2.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Cortex_A76AEx2.bp.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76AEx2.bp.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76AEx2.bp.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76AEx2.bp.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A76AEx2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76AEx2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76AEx2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76AEx2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76AEx2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A76AEx2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A76AEx2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A76AEx2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A76AEx2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A76AEx2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A76AEx2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A76AEx2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A76AEx2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A76AEx2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76AEx2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76AEx2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A76AEx2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A76AEx2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A76AEx2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A76AEx2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A76AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A76AEx2.bp.pl111_clcd_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_A76AEx2.bp.pl180_mci`

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A76AEx2.bp.ps2keyboard`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A76AEx2.bp.ps2mouse`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A76AEx2.bp.psram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76AEx2.bp.refcounter`

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A76AEx2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A76AEx2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76AEx2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76AEx2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A76AEx2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76AEx2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76AEx2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76AEx2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76AEx2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76AEx2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76AEx2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A76AEx2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A76AEx2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A76AEx2.bp.trusted_watchdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A76AEx2.bp.tzc_400

TrustZone Address Space Controller.
Type: `TZC_400`.

FVP_Base_Cortex_A76AEx2.bp.ve_sysregs

Type: `VE_SysRegs`.

FVP_Base_Cortex_A76AEx2.bp.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_Base_Cortex_A76AEx2.bp.virtio_net

VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A76AEx2.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A76AEx2.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A76AEx2.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A76AEx2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A76AEx2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A76AEx2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A76AEx2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A76AEx2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A76AEx2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx2.cluster0

ARM Cortex-A76AE Cluster CT model.
Type: `Cluster_ARM_Cortex-A76AE`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu0

ARM Cortex-A76AE CT model.
Type: `ARM_Cortex-A76AE`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu1

ARM Cortex-A76AE CT model.
Type: `ARM_Cortex-A76AE`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu1.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx2.cluster0.cpu1.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx2.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A76AEx2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A76AEx2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A76AEx2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A76AEx2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.69 FVP_Base_Cortex-A76AEx4

FVP_Base_Cortex-A76AEx4 contains the following instances:

FVP_Base_Cortex-A76AEx4 instances

FVP_Base_Cortex_A76AEx4
Base Platform Compute Subsystem for ARMCortexA76AEx4CT.
Type: FVP_Base_Cortex_A76AEx4.

FVP_Base_Cortex_A76AEx4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A76AEx4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A76AEx4.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A76AEx4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A76AEx4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A76AEx4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A76AEx4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A76AEx4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A76AEx4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A76AEx4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A76AEx4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A76AEx4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A76AEx4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76AEx4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76AEx4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76AEx4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76AEx4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A76AEx4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A76AEx4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A76AEx4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A76AEx4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A76AEx4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A76AEx4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A76AEx4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A76AEx4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A76AEx4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A76AEx4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A76AEx4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A76AEx4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A76AEx4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A76AEx4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A76AEx4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76AEx4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A76AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A76AEx4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A76AEx4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A76AEx4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A76AEx4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A76AEx4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A76AEx4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A76AEx4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76AEx4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76AEx4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A76AEx4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76AEx4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76AEx4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76AEx4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76AEx4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76AEx4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A76AEx4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A76AEx4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A76AEx4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76AEx4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A76AEx4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A76AEx4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A76AEx4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A76AEx4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A76AEx4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A76AEx4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A76AEx4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A76AEx4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A76AEx4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A76AEx4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A76AEx4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76AEx4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A76AEx4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76AEx4.cluster0

ARM Cortex-A76AE Cluster CT model.
Type: Cluster_ARM_Cortex-A76AE.

FVP_Base_Cortex_A76AEx4.cluster0.cpu0

ARM Cortex-A76AE CT model.
Type: ARM_Cortex-A76AE.

FVP_Base_Cortex_A76AEx4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A76AEx4.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu1
ARM Cortex-A76AE CT model.
Type: ARM_Cortex-A76AE.

FVP_Base_Cortex_A76AEx4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A76AEx4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu2
ARM Cortex-A76AE CT model.
Type: ARM_Cortex-A76AE.

FVP_Base_Cortex_A76AEx4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A76AEx4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76AEx4.cluster0.cpu3
ARM Cortex-A76AE CT model.
Type: `ARM_Cortex-A76AE`.

FVP_Base_Cortex_A76AEx4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76AEx4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx4.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76AEx4.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A76AEx4.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A76AEx4.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A76AEx4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A76AEx4.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.70 FVP_Base_Cortex-A76x1

FVP_Base_Cortex-A76x1 contains the following instances:

FVP_Base_Cortex-A76x1 instances

FVP_Base_Cortex_A76x1
Base Platform Compute Subsystem for ARMCortexA76x1CT.
Type: `FVP_Base_Cortex_A76x1`.

FVP_Base_Cortex_A76x1.bp
Peripherals and address map for the Base Platform.

Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A76x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A76x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A76x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A76x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A76x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A76x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A76x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A76x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x1.bp.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A76x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A76x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A76x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A76x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A76x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A76x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A76x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A76x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A76x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A76x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A76x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A76x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A76x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A76x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A76x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A76x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A76x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A76x1.bp.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A76x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A76x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A76x1.bp.pl111_clcd_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_A76x1.bp.pl180_mci`

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A76x1.bp.ps2keyboard`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A76x1.bp.ps2mouse`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A76x1.bp.psram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76x1.bp.refcounter`

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

`FVP_Base_Cortex_A76x1.bp.reset_or`

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A76x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A76x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A76x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76x1.bp.sram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A76x1.bp.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A76x1.bp.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A76x1.bp.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A76x1.bp.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A76x1.bp.trusted_key_storage`
Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

`FVP_Base_Cortex_A76x1.bp.trusted_nv_counter`
Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

`FVP_Base_Cortex_A76x1.bp.trusted_rng`
Random Number Generator unit.
Type: `RandomNumberGenerator`.

`FVP_Base_Cortex_A76x1.bp.trusted_watchdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_Base_Cortex_A76x1.bp.tzc_400`
TrustZone Address Space Controller.
Type: `TZC_400`.

`FVP_Base_Cortex_A76x1.bp.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_Base_Cortex_A76x1.bp.vedcc`
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

`FVP_Base_Cortex_A76x1.bp.virtio_net`
VirtioNet device over MMIO transport.
Type: `VirtioNetMMIO`.

FVP_Base_Cortex_A76x1.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A76x1.bp.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_Base_Cortex_A76x1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A76x1.bp.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_Base_Cortex_A76x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A76x1.bp.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_Base_Cortex_A76x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_Base_Cortex_A76x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A76x1.bp.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A76x1.cci400

Cache Coherent Interconnect for AXI4 ACE.

Type: CCI400.

FVP_Base_Cortex_A76x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: `ClockDivider`.

`FVP_Base_Cortex_A76x1.cluster0`

ARM Cortex-A76 Cluster CT model.
Type: `Cluster_ARM_Cortex-A76`.

`FVP_Base_Cortex_A76x1.cluster0.cpu0`

ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

`FVP_Base_Cortex_A76x1.cluster0.cpu0.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Cortex_A76x1.cluster0.cpu0.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A76x1.cluster0.cpu0.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A76x1.cluster0.cpu0.l2cache`

PV Cache.
Type: `PVCache`.

`FVP_Base_Cortex_A76x1.cluster0_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_A76x1.dapmemlogger`

Bus Logger.
Type: `PVBusLogger`.

`FVP_Base_Cortex_A76x1.elfloader`

ELF loader component.
Type: `ElfLoader`.

`FVP_Base_Cortex_A76x1.gic_distributor`

GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

`FVP_Base_Cortex_A76x1.pctl`

Base Platforms Power Controller.
Type: `Base_PowerController`.

7.71 FVP_Base_Cortex-A76x2

FVP_Base_Cortex-A76x2 contains the following instances:

FVP_Base_Cortex-A76x2 instances

FVP_Base_Cortex_A76x2

Base Platform Compute Subsystem for ARMCortexA76x2CT.
Type: `FVP_Base_Cortex_A76x2`.

FVP_Base_Cortex_A76x2.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A76x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A76x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A76x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A76x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A76x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A76x2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A76x2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x2.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A76x2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A76x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A76x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A76x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A76x2.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A76x2.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A76x2.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A76x2.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A76x2.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A76x2.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A76x2.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A76x2.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A76x2.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76x2.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A76x2.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A76x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A76x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A76x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A76x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A76x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A76x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A76x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A76x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A76x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A76x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A76x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A76x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A76x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76x2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76x2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76x2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A76x2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A76x2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A76x2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A76x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A76x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A76x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A76x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A76x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A76x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A76x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A76x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A76x2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A76x2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A76x2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A76x2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A76x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x2.cluster0

ARM Cortex-A76 Cluster CT model.
Type: `Cluster_ARM_Cortex-A76`.

FVP_Base_Cortex_A76x2.cluster0.cpu0

ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A76x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76x2.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x2.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x2.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x2.cluster0.cpu1

ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A76x2.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A76x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A76x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A76x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A76x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.72 FVP_Base_Cortex-A76x4

FVP_Base_Cortex-A76x4 contains the following instances:

FVP_Base_Cortex-A76x4 instances

FVP_Base_Cortex_A76x4
Base Platform Compute Subsystem for ARMCortexA76x4CT.
Type: FVP_Base_Cortex_A76x4.

FVP_Base_Cortex_A76x4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A76x4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A76x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A76x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A76x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A76x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A76x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A76x4.bp.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x4.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A76x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A76x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A76x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A76x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A76x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A76x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A76x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A76x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A76x4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A76x4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A76x4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A76x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A76x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A76x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A76x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76x4.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A76x4.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A76x4.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A76x4.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76x4.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A76x4.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A76x4.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_A76x4.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_A76x4.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A76x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A76x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A76x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A76x4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A76x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A76x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A76x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A76x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A76x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A76x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A76x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A76x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A76x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A76x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A76x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A76x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A76x4.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76x4.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A76x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A76x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A76x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A76x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A76x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A76x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A76x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A76x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A76x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A76x4.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A76x4.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A76x4.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A76x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A76x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A76x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A76x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A76x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A76x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A76x4.cluster0
ARM Cortex-A76 Cluster CT model.
Type: `Cluster_ARM_Cortex-A76`.

FVP_Base_Cortex_A76x4.cluster0.cpu0
ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A76x4.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76x4.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x4.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x4.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x4.cluster0.cpu1
ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A76x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x4.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A76x4.cluster0.cpu2
ARM Cortex-A76 CT model.
Type: `ARM_Cortex-A76`.

FVP_Base_Cortex_A76x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A76x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x4.cluster0.cpu3
ARM Cortex-A76 CT model.
Type: ARM_Cortex-A76.

FVP_Base_Cortex_A76x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A76x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A76x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A76x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A76x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A76x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A76x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.73 FVP_Base_Cortex-A77x1

FVP_Base_Cortex-A77x1 contains the following instances:

FVP_Base_Cortex-A77x1 instances

FVP_Base_Cortex_A77x1

Base Platform Compute Subsystem for ARMCortexA77x1CT.
Type: `FVP_Base_Cortex_A77x1`.

FVP_Base_Cortex_A77x1.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A77x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A77x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A77x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A77x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A77x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A77x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A77x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A77x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A77x1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A77x1.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A77x1.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A77x1.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A77x1.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A77x1.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A77x1.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A77x1.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A77x1.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A77x1.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A77x1.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A77x1.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A77x1.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A77x1.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A77x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A77x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A77x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A77x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A77x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A77x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A77x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A77x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A77x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A77x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A77x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A77x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A77x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A77x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A77x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A77x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A77x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A77x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A77x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A77x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A77x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A77x1.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A77x1.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A77x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A77x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A77x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A77x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A77x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A77x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A77x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A77x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A77x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A77x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A77x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A77x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A77x1.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A77x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x1.cluster0

ARM Cortex-A77 Cluster CT model.
Type: `Cluster_ARM_Cortex-A77`.

FVP_Base_Cortex_A77x1.cluster0.cpu0

ARM Cortex-A77 CT model.
Type: `ARM_Cortex-A77`.

FVP_Base_Cortex_A77x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A77x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A77x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A77x1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A77x1.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A77x1.dapmemlogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A77x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A77x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A77x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.74 FVP_Base_Cortex-A77x2

FVP_Base_Cortex-A77x2 contains the following instances:

FVP_Base_Cortex-A77x2 instances

FVP_Base_Cortex_A77x2

Base Platform Compute Subsystem for ARMCortexA77x2CT.
Type: FVP_Base_Cortex_A77x2.

FVP_Base_Cortex_A77x2.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A77x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A77x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A77x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A77x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A77x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A77x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A77x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A77x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A77x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A77x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A77x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A77x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A77x2.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A77x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A77x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A77x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A77x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A77x2.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A77x2.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A77x2.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A77x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A77x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A77x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A77x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A77x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A77x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A77x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A77x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A77x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A77x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A77x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A77x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A77x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A77x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A77x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A77x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A77x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A77x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A77x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A77x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A77x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A77x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A77x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A77x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A77x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A77x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A77x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A77x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A77x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A77x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A77x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A77x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A77x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A77x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A77x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A77x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A77x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A77x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A77x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A77x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A77x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A77x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A77x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A77x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A77x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A77x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x2.cluster0

ARM Cortex-A77 Cluster CT model.
Type: Cluster_ARM_Cortex-A77.

FVP_Base_Cortex_A77x2.cluster0.cpu0

ARM Cortex-A77 CT model.
Type: ARM_Cortex-A77.

FVP_Base_Cortex_A77x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A77x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x2.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x2.cluster0.cpu1
ARM Cortex-A77 CT model.
Type: ARM_Cortex-A77.

FVP_Base_Cortex_A77x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A77x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A77x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A77x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A77x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A77x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.75 FVP_Base_Cortex-A77x4

FVP_Base_Cortex-A77x4 contains the following instances:

FVP_Base_Cortex-A77x4 instances

FVP_Base_Cortex_A77x4

Base Platform Compute Subsystem for ARMCortexA77x4CT.
Type: `FVP_Base_Cortex_A77x4`.

FVP_Base_Cortex_A77x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A77x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A77x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A77x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A77x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A77x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A77x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A77x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A77x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A77x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A77x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A77x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A77x4.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A77x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A77x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A77x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A77x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A77x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A77x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A77x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A77x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A77x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A77x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A77x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A77x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A77x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A77x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A77x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A77x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A77x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A77x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A77x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A77x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A77x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A77x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A77x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A77x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A77x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A77x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A77x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A77x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A77x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A77x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A77x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A77x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A77x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A77x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A77x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A77x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A77x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A77x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A77x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A77x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A77x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A77x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A77x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A77x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A77x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A77x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A77x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A77x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A77x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A77x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A77x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A77x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A77x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A77x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A77x4.cluster0

ARM Cortex-A77 Cluster CT model.
Type: `Cluster_ARM_Cortex-A77`.

FVP_Base_Cortex_A77x4.cluster0.cpu0

ARM Cortex-A77 CT model.
Type: `ARM_Cortex-A77`.

FVP_Base_Cortex_A77x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A77x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A77x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A77x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A77x4.cluster0.cpu1

ARM Cortex-A77 CT model.
Type: `ARM_Cortex-A77`.

FVP_Base_Cortex_A77x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A77x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu2
ARM Cortex-A77 CT model.
Type: ARM_Cortex-A77.

FVP_Base_Cortex_A77x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A77x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu3
ARM Cortex-A77 CT model.
Type: ARM_Cortex-A77.

FVP_Base_Cortex_A77x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A77x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A77x4.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A77x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A77x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A77x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A77x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.76 FVP_Base_Cortex-A78AEx2

FVP_Base_Cortex-A78AEx2 contains the following instances:

FVP_Base_Cortex-A78AEx2 instances

FVP_Base_Cortex_A78AEx2

Base Platform Compute Subsystem for ARMCortexA78AEx2CT.
Type: FVP_Base_Cortex_A78AEx2.

FVP_Base_Cortex_A78AEx2.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A78AEx2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A78AEx2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78AEx2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78AEx2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A78AEx2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78AEx2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78AEx2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A78AEx2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A78AEx2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78AEx2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78AEx2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78AEx2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78AEx2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78AEx2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78AEx2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78AEx2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78AEx2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78AEx2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78AEx2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78AEx2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78AEx2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78AEx2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A78AEx2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A78AEx2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A78AEx2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78AEx2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78AEx2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A78AEx2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A78AEx2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78AEx2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78AEx2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A78AEx2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A78AEx2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A78AEx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78AEx2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A78AEx2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A78AEx2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A78AEx2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A78AEx2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A78AEx2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A78AEx2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78AEx2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78AEx2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78AEx2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78AEx2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78AEx2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78AEx2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78AEx2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78AEx2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78AEx2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A78AEx2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A78AEx2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A78AEx2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78AEx2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78AEx2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78AEx2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78AEx2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78AEx2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78AEx2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78AEx2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A78AEx2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78AEx2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A78AEx2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78AEx2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78AEx2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A78AEx2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx2.cluster0

ARM Cortex-A78AE Cluster CT model.
Type: Cluster_ARM_Cortex-A78AE.

FVP_Base_Cortex_A78AEx2.cluster0.cpu0

ARM Cortex-A78AE CT model.
Type: ARM_Cortex-A78AE.

FVP_Base_Cortex_A78AEx2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78AEx2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx2.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx2.cluster0.cpu1
ARM Cortex-A78AE CT model.
Type: ARM_Cortex-A78AE.

FVP_Base_Cortex_A78AEx2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78AEx2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A78AEx2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A78AEx2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78AEx2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78AEx2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.77 FVP_Base_Cortex-A78AEx4

FVP_Base_Cortex-A78AEx4 contains the following instances:

FVP_Base_Cortex-A78AEx4 instances

FVP_Base_Cortex_A78AEx4

Base Platform Compute Subsystem for ARMCortexA78AEx4CT.
Type: `FVP_Base_Cortex_A78AEx4`.

FVP_Base_Cortex_A78AEx4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A78AEx4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78AEx4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78AEx4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78AEx4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78AEx4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78AEx4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78AEx4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A78AEx4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A78AEx4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78AEx4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78AEx4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78AEx4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78AEx4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78AEx4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78AEx4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78AEx4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78AEx4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78AEx4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78AEx4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78AEx4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A78AEx4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A78AEx4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A78AEx4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A78AEx4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A78AEx4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A78AEx4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A78AEx4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A78AEx4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78AEx4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78AEx4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78AEx4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78AEx4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78AEx4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A78AEx4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A78AEx4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78AEx4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78AEx4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A78AEx4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A78AEx4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78AEx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78AEx4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A78AEx4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A78AEx4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A78AEx4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A78AEx4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A78AEx4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A78AEx4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A78AEx4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A78AEx4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Cortex_A78AEx4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Cortex_A78AEx4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Cortex_A78AEx4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78AEx4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78AEx4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78AEx4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78AEx4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A78AEx4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A78AEx4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A78AEx4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78AEx4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78AEx4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78AEx4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78AEx4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78AEx4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78AEx4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78AEx4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A78AEx4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78AEx4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A78AEx4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78AEx4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78AEx4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78AEx4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78AEx4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A78AEx4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78AEx4.cluster0

ARM Cortex-A78AE Cluster CT model.
Type: `Cluster_ARM_Cortex-A78AE`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu0

ARM Cortex-A78AE CT model.
Type: `ARM_Cortex-A78AE`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu1

ARM Cortex-A78AE CT model.
Type: `ARM_Cortex-A78AE`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78AEx4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu2
ARM Cortex-A78AE CT model.
Type: ARM_Cortex-A78AE.

FVP_Base_Cortex_A78AEx4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78AEx4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu3
ARM Cortex-A78AE CT model.
Type: ARM_Cortex-A78AE.

FVP_Base_Cortex_A78AEx4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78AEx4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78AEx4.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A78AEx4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A78AEx4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78AEx4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78AEx4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.78 FVP_Base_Cortex-A78C

FVP_Base_Cortex-A78C contains the following instances:

FVP_Base_Cortex-A78C instances

FVP_Base_Cortex_A78C

Base Platform Compute Subsystem for ARMCortexA78CCT.
Type: FVP_Base_Cortex_A78C.

FVP_Base_Cortex_A78C.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A78C.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A78C.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78C.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78C.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A78C.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78C.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78C.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A78C.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A78C.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78C.bp.clock300MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78C.bp.clock32KHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78C.bp.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78C.bp.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78C.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78C.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78C.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78C.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78C.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78C.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78C.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78C.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78C.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78C.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78C.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78C.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78C.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78C.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78C.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78C.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78C.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A78C.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A78C.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A78C.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78C.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78C.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78C.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78C.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78C.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78C.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A78C.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A78C.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78C.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78C.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A78C.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A78C.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78C.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78C.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_A78C.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78C.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A78C.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A78C.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A78C.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A78C.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A78C.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A78C.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78C.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78C.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78C.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78C.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78C.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78C.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78C.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78C.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78C.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A78C.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A78C.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A78C.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78C.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78C.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78C.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78C.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78C.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78C.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78C.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A78C.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78C.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A78C.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78C.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78C.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78C.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A78C.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78C.cluster0

ARM Cortex-A78C Cluster CT model.
Type: Cluster_ARM_Cortex-A78C.

FVP_Base_Cortex_A78C.cluster0.cpu0

ARM Cortex-A78C CT model.
Type: ARM_Cortex-A78C.

FVP_Base_Cortex_A78C.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78C.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu1
ARM Cortex-A78C CT model.
Type: ARM_Cortex-A78C.

FVP_Base_Cortex_A78C.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78C.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu2
ARM Cortex-A78C CT model.
Type: ARM_Cortex-A78C.

FVP_Base_Cortex_A78C.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78C.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78C.cluster0.cpu3
ARM Cortex-A78C CT model.
Type: `ARM_Cortex-A78C`.

FVP_Base_Cortex_A78C.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78C.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78C.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78C.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78C.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_A78C.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A78C.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Cortex_A78C.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Cortex_A78C.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.79 FVP_Base_Cortex-A78Cx1

FVP_Base_Cortex-A78Cx1 contains the following instances:

FVP_Base_Cortex-A78Cx1 instances

FVP_Base_Cortex_A78Cx1
Base Platform Compute Subsystem for ARMCortexA78Cx1CT.
Type: `FVP_Base_Cortex_A78Cx1`.

FVP_Base_Cortex_A78Cx1.bp
Peripherals and address map for the Base Platform.

Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A78Cx1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78Cx1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78Cx1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A78Cx1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A78Cx1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A78Cx1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78Cx1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78Cx1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx1.bp.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Cortex_A78Cx1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A78Cx1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78Cx1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78Cx1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78Cx1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78Cx1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78Cx1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78Cx1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A78Cx1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A78Cx1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A78Cx1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A78Cx1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A78Cx1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A78Cx1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A78Cx1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A78Cx1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_A78Cx1.bp.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A78Cx1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A78Cx1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A78Cx1.bp.pl111_clcd_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A78Cx1.bp.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Cortex_A78Cx1.bp.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Cortex_A78Cx1.bp.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Cortex_A78Cx1.bp.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A78Cx1.bp.refcounter`
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

`FVP_Base_Cortex_A78Cx1.bp.reset_or`
Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A78Cx1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78Cx1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78Cx1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78Cx1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78Cx1.bp.sram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_A78Cx1.bp.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A78Cx1.bp.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A78Cx1.bp.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A78Cx1.bp.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Cortex_A78Cx1.bp.trusted_key_storage`
Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

`FVP_Base_Cortex_A78Cx1.bp.trusted_nv_counter`
Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

`FVP_Base_Cortex_A78Cx1.bp.trusted_rng`
Random Number Generator unit.
Type: `RandomNumberGenerator`.

`FVP_Base_Cortex_A78Cx1.bp.trusted_watchdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_Base_Cortex_A78Cx1.bp.tzc_400`
TrustZone Address Space Controller.
Type: `TZC_400`.

`FVP_Base_Cortex_A78Cx1.bp.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_Base_Cortex_A78Cx1.bp.vedcc`
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

`FVP_Base_Cortex_A78Cx1.bp.virtio_net`
VirtioNet device over MMIO transport.
Type: `VirtioNetMMIO`.

FVP_Base_Cortex_A78Cx1.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx1.bp.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_Base_Cortex_A78Cx1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx1.bp.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_Base_Cortex_A78Cx1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx1.bp.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_Base_Cortex_A78Cx1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_Base_Cortex_A78Cx1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.bp.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Cortex_A78Cx1.cci400

Cache Coherent Interconnect for AXI4 ACE.

Type: CCI400.

FVP_Base_Cortex_A78Cx1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A78Cx1.cluster0

ARM Cortex-A78C Cluster CT model.
Type: Cluster_ARM_Cortex-A78C.

FVP_Base_Cortex_A78Cx1.cluster0.cpu0

ARM Cortex-A78C CT model.
Type: ARM_Cortex-A78C.

FVP_Base_Cortex_A78Cx1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78Cx1.cluster0.cpu0.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx1.cluster0.cpu0.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx1.cluster0.cpu0.l2cache

PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx1.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A78Cx1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78Cx1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78Cx1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.80 FVP_Base_Cortex-A78Cx2

FVP_Base_Cortex-A78Cx2 contains the following instances:

FVP_Base_Cortex-A78Cx2 instances

FVP_Base_Cortex_A78Cx2

Base Platform Compute Subsystem for ARMCortexA78Cx2CT.
Type: `FVP_Base_Cortex_A78Cx2`.

FVP_Base_Cortex_A78Cx2.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A78Cx2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78Cx2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78Cx2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A78Cx2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A78Cx2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78Cx2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78Cx2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx2.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78Cx2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78Cx2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78Cx2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78Cx2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A78Cx2.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A78Cx2.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A78Cx2.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A78Cx2.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A78Cx2.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A78Cx2.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A78Cx2.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A78Cx2.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78Cx2.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78Cx2.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78Cx2.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A78Cx2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A78Cx2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78Cx2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78Cx2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A78Cx2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A78Cx2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78Cx2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78Cx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78Cx2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78Cx2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A78Cx2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A78Cx2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A78Cx2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A78Cx2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A78Cx2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78Cx2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78Cx2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78Cx2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx2.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78Cx2.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78Cx2.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78Cx2.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78Cx2.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A78Cx2.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A78Cx2.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A78Cx2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78Cx2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78Cx2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78Cx2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78Cx2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78Cx2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78Cx2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78Cx2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A78Cx2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78Cx2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A78Cx2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78Cx2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78Cx2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A78Cx2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx2.cluster0

ARM Cortex-A78C Cluster CT model.
Type: `Cluster_ARM_Cortex-A78C`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu0

ARM Cortex-A78C CT model.
Type: `ARM_Cortex-A78C`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu1

ARM Cortex-A78C CT model.
Type: `ARM_Cortex-A78C`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78Cx2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A78Cx2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A78Cx2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78Cx2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78Cx2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.81 FVP_Base_Cortex-A78Cx4

FVP_Base_Cortex-A78Cx4 contains the following instances:

FVP_Base_Cortex-A78Cx4 instances

FVP_Base_Cortex_A78Cx4
Base Platform Compute Subsystem for ARMCortexA78Cx4CT.
Type: FVP_Base_Cortex_A78Cx4.

FVP_Base_Cortex_A78Cx4.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A78Cx4.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A78Cx4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78Cx4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78Cx4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A78Cx4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A78Cx4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78Cx4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78Cx4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78Cx4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78Cx4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78Cx4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78Cx4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78Cx4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78Cx4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78Cx4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78Cx4.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx4.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_A78Cx4.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_A78Cx4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78Cx4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78Cx4.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_A78Cx4.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_A78Cx4.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A78Cx4.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78Cx4.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_A78Cx4.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78Cx4.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_A78Cx4.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_A78Cx4.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A78Cx4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A78Cx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A78Cx4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A78Cx4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_A78Cx4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_A78Cx4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_A78Cx4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_A78Cx4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_A78Cx4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_A78Cx4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78Cx4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78Cx4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78Cx4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78Cx4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78Cx4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78Cx4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78Cx4.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78Cx4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78Cx4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78Cx4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A78Cx4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A78Cx4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A78Cx4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78Cx4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78Cx4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78Cx4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78Cx4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78Cx4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78Cx4.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78Cx4.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx4.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78Cx4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A78Cx4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78Cx4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78Cx4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78Cx4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A78Cx4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78Cx4.cluster0
ARM Cortex-A78C Cluster CT model.
Type: `Cluster_ARM_Cortex-A78C`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu0
ARM Cortex-A78C CT model.
Type: `ARM_Cortex-A78C`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu1
ARM Cortex-A78C CT model.
Type: `ARM_Cortex-A78C`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu2
ARM Cortex-A78C CT model.
Type: `ARM_Cortex-A78C`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78Cx4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx4.cluster0.cpu3
ARM Cortex-A78C CT model.
Type: ARM_Cortex-A78C.

FVP_Base_Cortex_A78Cx4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78Cx4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78Cx4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A78Cx4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A78Cx4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78Cx4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78Cx4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.82 FVP_Base_Cortex-A78x1

FVP_Base_Cortex-A78x1 contains the following instances:

FVP_Base_Cortex-A78x1 instances

FVP_Base_Cortex_A78x1

Base Platform Compute Subsystem for ARMCortexA78x1CT.
Type: `FVP_Base_Cortex_A78x1`.

FVP_Base_Cortex_A78x1.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A78x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A78x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A78x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78x1.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78x1.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x1.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x1.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x1.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78x1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A78x1.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A78x1.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A78x1.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A78x1.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A78x1.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A78x1.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A78x1.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A78x1.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78x1.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78x1.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78x1.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A78x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A78x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A78x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A78x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A78x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A78x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A78x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A78x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A78x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A78x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A78x1.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A78x1.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A78x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A78x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A78x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78x1.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A78x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x1.cluster0

ARM Cortex-A78 Cluster CT model.
Type: `Cluster_ARM_Cortex-A78`.

FVP_Base_Cortex_A78x1.cluster0.cpu0

ARM Cortex-A78 CT model.
Type: `ARM_Cortex-A78`.

FVP_Base_Cortex_A78x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78x1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78x1.cluster0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A78x1.dapmemlogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_A78x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.83 FVP_Base_Cortex-A78x2

FVP_Base_Cortex-A78x2 contains the following instances:

FVP_Base_Cortex-A78x2 instances

FVP_Base_Cortex_A78x2

Base Platform Compute Subsystem for ARMCortexA78x2CT.
Type: FVP_Base_Cortex_A78x2.

FVP_Base_Cortex_A78x2.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_A78x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A78x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Cortex_A78x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_A78x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_A78x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_A78x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_A78x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_A78x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78x2.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A78x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A78x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A78x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_A78x2.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Cortex_A78x2.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Cortex_A78x2.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Cortex_A78x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Cortex_A78x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Cortex_A78x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A78x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A78x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A78x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Cortex_A78x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Cortex_A78x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Cortex_A78x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A78x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Cortex_A78x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Cortex_A78x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Cortex_A78x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_A78x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_A78x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_A78x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A78x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A78x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A78x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A78x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A78x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A78x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_A78x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_A78x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_A78x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_A78x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A78x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_A78x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_A78x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x2.cluster0

ARM Cortex-A78 Cluster CT model.
Type: Cluster_ARM_Cortex-A78.

FVP_Base_Cortex_A78x2.cluster0.cpu0

ARM Cortex-A78 CT model.
Type: ARM_Cortex-A78.

FVP_Base_Cortex_A78x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x2.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x2.cluster0.cpu1
ARM Cortex-A78 CT model.
Type: ARM_Cortex-A78.

FVP_Base_Cortex_A78x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_A78x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A78x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.84 FVP_Base_Cortex-A78x4

FVP_Base_Cortex-A78x4 contains the following instances:

FVP_Base_Cortex-A78x4 instances

FVP_Base_Cortex_A78x4

Base Platform Compute Subsystem for ARMCortexA78x4CT.
Type: `FVP_Base_Cortex_A78x4`.

FVP_Base_Cortex_A78x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Cortex_A78x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_A78x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_A78x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_A78x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_A78x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Cortex_A78x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_A78x4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_A78x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_A78x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_A78x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_A78x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_A78x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_A78x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_A78x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_A78x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_A78x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_A78x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_A78x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_A78x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_A78x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_A78x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_A78x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_A78x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_A78x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_A78x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_A78x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_A78x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_A78x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_A78x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_A78x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_A78x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_A78x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_A78x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_A78x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_A78x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_A78x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_A78x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Cortex_A78x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Cortex_A78x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_A78x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_A78x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_A78x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_A78x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_A78x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_A78x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_A78x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_A78x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_A78x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_A78x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_A78x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_A78x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_A78x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_A78x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_A78x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_A78x4.cluster0

ARM Cortex-A78 Cluster CT model.
Type: `Cluster_ARM_Cortex-A78`.

FVP_Base_Cortex_A78x4.cluster0.cpu0

ARM Cortex-A78 CT model.
Type: `ARM_Cortex-A78`.

FVP_Base_Cortex_A78x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_A78x4.cluster0.cpu1

ARM Cortex-A78 CT model.
Type: `ARM_Cortex-A78`.

FVP_Base_Cortex_A78x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_A78x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu2
ARM Cortex-A78 CT model.
Type: ARM_Cortex-A78.

FVP_Base_Cortex_A78x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu3
ARM Cortex-A78 CT model.
Type: ARM_Cortex-A78.

FVP_Base_Cortex_A78x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_A78x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_A78x4.cluster0_labeller

Type: Labeller.

FVP_Base_Cortex_A78x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_A78x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_A78x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_A78x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.85 FVP_Base_Cortex-X1x1

FVP_Base_Cortex-X1x1 contains the following instances:

FVP_Base_Cortex-X1x1 instances

FVP_Base_Cortex_X1x1

Base Platform Compute Subsystem for ARMCortexX1x1CT.
Type: FVP_Base_Cortex_X1x1.

FVP_Base_Cortex_X1x1.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_X1x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X1x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X1x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_X1x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_X1x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x1.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x1.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x1.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x1.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x1.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x1.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x1.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X1x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X1x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X1x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_X1x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_X1x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_X1x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X1x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X1x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X1x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X1x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_X1x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_X1x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_X1x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_X1x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_X1x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X1x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X1x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_X1x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_X1x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X1x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X1x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_X1x1.bp.pl111_clcd.pl111x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X1x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_X1x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_X1x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_X1x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_X1x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_X1x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_X1x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_X1x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X1x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_X1x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_X1x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_X1x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_X1x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X1x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_X1x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_X1x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_X1x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_X1x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_X1x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_X1x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_X1x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_X1x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_X1x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_X1x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_X1x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_X1x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x1.cluster0

ARM Cortex-X1 Cluster CT model.
Type: Cluster_ARM_Cortex-X1.

FVP_Base_Cortex_X1x1.cluster0.cpu0

ARM Cortex-X1 CT model.
Type: ARM_Cortex-X1.

FVP_Base_Cortex_X1x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X1x1.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X1x1.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X1x1.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X1x1.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_X1x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_X1x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_X1x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_X1x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.86 FVP_Base_Cortex-X1x2

FVP_Base_Cortex-X1x2 contains the following instances:

FVP_Base_Cortex-X1x2 instances

FVP_Base_Cortex_X1x2
Base Platform Compute Subsystem for ARMCortexX1x2CT.
Type: FVP_Base_Cortex_X1x2.

FVP_Base_Cortex_X1x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_X1x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X1x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X1x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X1x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_X1x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X1x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X1x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_X1x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_X1x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X1x2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X1x2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_X1x2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_X1x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_X1x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X1x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X1x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X1x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X1x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_X1x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_X1x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_X1x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x2.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_X1x2.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_X1x2.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_X1x2.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x2.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_X1x2.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x2.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_X1x2.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_X1x2.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_X1x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_X1x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_X1x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_X1x2.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_X1x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_X1x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_X1x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_X1x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X1x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_X1x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_X1x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X1x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_X1x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X1x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_X1x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X1x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_X1x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_X1x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_X1x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X1x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_X1x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_X1x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_X1x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_X1x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_X1x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_X1x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_X1x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_X1x2.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_X1x2.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_X1x2.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_X1x2.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.bp.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x2.cci400
Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_X1x2.clockdivider0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x2.cluster0
ARM Cortex-X1 Cluster CT model.
Type: `Cluster_ARM_Cortex-X1`.

FVP_Base_Cortex_X1x2.cluster0.cpu0
ARM Cortex-X1 CT model.
Type: `ARM_Cortex-X1`.

FVP_Base_Cortex_X1x2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X1x2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x2.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x2.cluster0.cpu1
ARM Cortex-X1 CT model.
Type: `ARM_Cortex-X1`.

FVP_Base_Cortex_X1x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X1x2.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x2.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x2.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_X1x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_X1x2.elfloader
ELF loader component.

Type: ElfLoader.

FVP_Base_Cortex_X1x2.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_X1x2.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.87 FVP_Base_Cortex-X1x4

FVP_Base_Cortex-X1x4 contains the following instances:

FVP_Base_Cortex-X1x4 instances

FVP_Base_Cortex_X1x4

Base Platform Compute Subsystem for ARMCortexX1x4CT.
Type: FVP_Base_Cortex_X1x4.

FVP_Base_Cortex_X1x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_X1x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X1x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X1x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X1x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_X1x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_X1x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x4.bp.clock32KHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x4.bp.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x4.bp.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x4.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x4.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X1x4.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_X1x4.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Cortex_X1x4.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X1x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X1x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X1x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_X1x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_X1x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_X1x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_X1x4.bp.hdlcd0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_X1x4.bp.hdlcd0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_X1x4.bp.hdlcd0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_X1x4.bp.hdlcd0_labeller`

Type: `Labeller`.

`FVP_Base_Cortex_X1x4.bp.hostbridge`

Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_X1x4.bp.mmc`

Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_X1x4.bp.nontrustedrom`

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_X1x4.bp.nontrustedromloader`

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_X1x4.bp.pl011_uart0`

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_X1x4.bp.pl011_uart0.clk_divider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X1x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_X1x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_X1x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X1x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X1x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X1x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_X1x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_X1x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X1x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X1x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X1x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X1x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_X1x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).

Type: PL180_MCI.

FVP_Base_Cortex_X1x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_X1x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_X1x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_X1x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_X1x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X1x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X1x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X1x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_X1x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X1x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_X1x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X1x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X1x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X1x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X1x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X1x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_X1x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_X1x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_X1x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X1x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_X1x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_X1x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_X1x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_X1x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_X1x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_X1x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_X1x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_X1x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_X1x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_X1x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_X1x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X1x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_X1x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X1x4.cluster0

ARM Cortex-X1 Cluster CT model.
Type: `Cluster_ARM_Cortex-X1`.

FVP_Base_Cortex_X1x4.cluster0.cpu0

ARM Cortex-X1 CT model.
Type: `ARM_Cortex-X1`.

FVP_Base_Cortex_X1x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X1x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu1
ARM Cortex-X1 CT model.
Type: `ARM_Cortex-X1`.

FVP_Base_Cortex_X1x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X1x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu2
ARM Cortex-X1 CT model.
Type: `ARM_Cortex-X1`.

FVP_Base_Cortex_X1x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X1x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu2.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu3
ARM Cortex-X1 CT model.
Type: `ARM_Cortex-X1`.

FVP_Base_Cortex_X1x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X1x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X1x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X1x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X1x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_X1x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_X1x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_X1x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_X1x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.88 FVP_Base_Cortex-X2

FVP_Base_Cortex-X2 contains the following instances:

FVP_Base_Cortex-X2 instances

FVP_Base_Cortex_X2
Base Platform Compute Subsystem for ARMCortexX2CT.
Type: FVP_Base_Cortex_X2.

FVP_Base_Cortex_X2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_X2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X2.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_X2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_X2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_X2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_X2.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_X2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_X2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_X2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_X2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_X2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_X2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_X2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_X2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_X2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_X2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Cortex_X2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_X2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_X2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_X2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_X2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_X2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_X2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_X2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_X2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_X2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_X2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_X2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_X2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_X2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_X2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_X2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_X2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_X2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_X2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_X2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_X2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_X2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_X2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_X2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2.cluster0

ARM Cortex-X2 Cluster CT model.
Type: Cluster_ARM_Cortex-X2.

FVP_Base_Cortex_X2.cluster0.cpu0

ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu1
ARM Cortex-X2 CT model.
Type: `ARM_Cortex-X2`.

FVP_Base_Cortex_X2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X2.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu2
ARM Cortex-X2 CT model.
Type: `ARM_Cortex-X2`.

FVP_Base_Cortex_X2.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X2.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu3
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu4
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2.cluster0.cpu4.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2.cluster0.cpu4.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu4.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu4.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu5
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2.cluster0.cpu5.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2.cluster0.cpu5.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu5.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu5.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu6
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2.cluster0.cpu6.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2.cluster0.cpu6.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu6.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu6.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu7
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2.cluster0.cpu7.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2.cluster0.cpu7.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu7.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0.cpu7.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_X2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_X2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_X2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_X2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.89 FVP_Base_Cortex-X2x1

FVP_Base_Cortex-X2x1 contains the following instances:

FVP_Base_Cortex-X2x1 instances

FVP_Base_Cortex_X2x1
Base Platform Compute Subsystem for ARMCortexX2x1CT.
Type: FVP_Base_Cortex_X2x1.

FVP_Base_Cortex_X2x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_X2x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X2x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X2x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_X2x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_X2x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_X2x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_X2x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_X2x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X2x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X2x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X2x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X2x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_X2x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_X2x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_X2x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_X2x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_X2x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X2x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X2x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_X2x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_X2x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X2x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X2x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Cortex_X2x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X2x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_X2x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Cortex_X2x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_X2x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_X2x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_X2x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_X2x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_X2x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_X2x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_X2x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_X2x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_X2x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_X2x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_X2x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_X2x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_X2x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_X2x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_X2x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_X2x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_X2x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_X2x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_X2x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_X2x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_X2x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x1.cluster0

ARM Cortex-X2 Cluster CT model.
Type: Cluster_ARM_Cortex-X2.

FVP_Base_Cortex_X2x1.cluster0.cpu0

ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2x1.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x1.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x1.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x1.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_X2x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_X2x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_X2x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_X2x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.90 FVP_Base_Cortex-X2x2

FVP_Base_Cortex-X2x2 contains the following instances:

FVP_Base_Cortex-X2x2 instances

FVP_Base_Cortex_X2x2
Base Platform Compute Subsystem for ARMCortexX2x2CT.
Type: FVP_Base_Cortex_X2x2.

FVP_Base_Cortex_X2x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_X2x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X2x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Cortex_X2x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Cortex_X2x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Cortex_X2x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Cortex_X2x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_X2x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_X2x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_X2x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X2x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X2x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X2x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X2x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Cortex_X2x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Cortex_X2x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Cortex_X2x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X2x2.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Cortex_X2x2.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Cortex_X2x2.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_X2x2.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X2x2.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Cortex_X2x2.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Cortex_X2x2.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Cortex_X2x2.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Cortex_X2x2.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Cortex_X2x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Cortex_X2x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Cortex_X2x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Cortex_X2x2.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Cortex_X2x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Cortex_X2x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Cortex_X2x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Cortex_X2x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X2x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Cortex_X2x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Cortex_X2x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X2x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_X2x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_X2x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Cortex_X2x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Cortex_X2x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_X2x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_X2x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_X2x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_X2x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_X2x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_X2x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_X2x2.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_X2x2.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Cortex_X2x2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_X2x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Cortex_X2x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_X2x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_X2x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Cortex_X2x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x2.cluster0
ARM Cortex-X2 Cluster CT model.
Type: `Cluster_ARM_Cortex-X2`.

FVP_Base_Cortex_X2x2.cluster0.cpu0
ARM Cortex-X2 CT model.
Type: `ARM_Cortex-X2`.

FVP_Base_Cortex_X2x2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X2x2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x2.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x2.cluster0.cpu1
ARM Cortex-X2 CT model.
Type: `ARM_Cortex-X2`.

FVP_Base_Cortex_X2x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X2x2.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x2.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x2.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Cortex_X2x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Cortex_X2x2.elfloader
ELF loader component.

Type: ElfLoader.

FVP_Base_Cortex_X2x2.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_X2x2.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.91 FVP_Base_Cortex-X2x4

FVP_Base_Cortex-X2x4 contains the following instances:

FVP_Base_Cortex-X2x4 instances

FVP_Base_Cortex_X2x4

Base Platform Compute Subsystem for ARMCortexX2x4CT.
Type: FVP_Base_Cortex_X2x4.

FVP_Base_Cortex_X2x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Cortex_X2x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X2x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Cortex_X2x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Cortex_X2x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Cortex_X2x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Cortex_X2x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Cortex_X2x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Cortex_X2x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Cortex_X2x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Cortex_X2x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Cortex_X2x4.bp.hdlcd0.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Cortex_X2x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Cortex_X2x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Cortex_X2x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Cortex_X2x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Cortex_X2x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Cortex_X2x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Cortex_X2x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Cortex_X2x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Cortex_X2x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Cortex_X2x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Cortex_X2x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Cortex_X2x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X2x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Cortex_X2x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Cortex_X2x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Cortex_X2x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Cortex_X2x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Cortex_X2x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Cortex_X2x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Cortex_X2x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Cortex_X2x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Cortex_X2x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).

Type: PL180_MCI.

FVP_Base_Cortex_X2x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Cortex_X2x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Cortex_X2x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Cortex_X2x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Cortex_X2x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Cortex_X2x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Cortex_X2x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Cortex_X2x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Cortex_X2x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Cortex_X2x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X2x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X2x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X2x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X2x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Cortex_X2x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Cortex_X2x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Cortex_X2x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Cortex_X2x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Cortex_X2x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Cortex_X2x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Cortex_X2x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Cortex_X2x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Cortex_X2x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Cortex_X2x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Cortex_X2x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Cortex_X2x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Cortex_X2x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Cortex_X2x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Cortex_X2x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Cortex_X2x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Cortex_X2x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Cortex_X2x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Cortex_X2x4.cluster0

ARM Cortex-X2 Cluster CT model.
Type: `Cluster_ARM_Cortex-X2`.

FVP_Base_Cortex_X2x4.cluster0.cpu0

ARM Cortex-X2 CT model.
Type: `ARM_Cortex-X2`.

FVP_Base_Cortex_X2x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Cortex_X2x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Cortex_X2x4.cluster0.cpu1
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu2
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu3
ARM Cortex-X2 CT model.
Type: ARM_Cortex-X2.

FVP_Base_Cortex_X2x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Cortex_X2x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Cortex_X2x4.cluster0_labeller
Type: Labeller.

FVP_Base_Cortex_X2x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Cortex_X2x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Cortex_X2x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Cortex_X2x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.92 FVP_Base_Neoverse-E1x1

FVP_Base_Neoverse-E1x1 contains the following instances:

FVP_Base_Neoverse-E1x1 instances

FVP_Base_Neoverse_E1x1
Base Platform Compute Subsystem for ARMNeoverseE1x1CT.
Type: FVP_Base_Neoverse_E1x1.

FVP_Base_Neoverse_E1x1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_E1x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_E1x1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_E1x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_E1x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Neoverse_E1x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_E1x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_E1x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Neoverse_E1x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Neoverse_E1x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_E1x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_E1x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_E1x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_E1x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_E1x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_E1x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_E1x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_E1x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_E1x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_E1x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_E1x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_E1x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_E1x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_E1x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_E1x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_E1x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_E1x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_E1x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Neoverse_E1x1.bp.pl111_clcd.pl111x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_E1x1.bp.pl111_clcd.pl111x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_E1x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Neoverse_E1x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Neoverse_E1x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Neoverse_E1x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Neoverse_E1x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Neoverse_E1x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_E1x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_E1x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_E1x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x1.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Neoverse_E1x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_E1x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_E1x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_E1x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_E1x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_E1x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_E1x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_E1x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_E1x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_E1x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_E1x1.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_E1x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_E1x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_E1x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_E1x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x1.cluster0

ARM Neoverse-E1 Cluster CT model.
Type: Cluster_ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_E1x1.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x1.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x1.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x1.cluster0.cpu0.thread0
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x1.cluster0.cpu0.thread1
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x1.cluster0_labeller
Type: Labeller.

FVP_Base_Neoverse_E1x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_E1x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_E1x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_E1x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.93 FVP_Base_Neoverse-E1x2

FVP_Base_Neoverse-E1x2 contains the following instances:

FVP_Base_Neoverse-E1x2 instances

FVP_Base_Neoverse_E1x2
Base Platform Compute Subsystem for ARMNeoverseE1x2CT.
Type: FVP_Base_Neoverse_E1x2.

FVP_Base_Neoverse_E1x2.bp
Peripherals and address map for the Base Platform.

Type: `BasePlatformPeripherals`.

`FVP_Base_Neoverse_E1x2.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Neoverse_E1x2.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x2.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x2.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Neoverse_E1x2.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Neoverse_E1x2.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Neoverse_E1x2.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x2.bp.Timer_2_3.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x2.bp.Timer_2_3.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Neoverse_E1x2.bp.Timer_2_3.counter1`

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Neoverse_E1x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Neoverse_E1x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Neoverse_E1x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_E1x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_E1x2.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_E1x2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Neoverse_E1x2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Neoverse_E1x2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_E1x2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_E1x2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_E1x2.bp.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Neoverse_E1x2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_E1x2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_E1x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_E1x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_E1x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_E1x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_E1x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_E1x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_E1x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_E1x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Neoverse_E1x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Neoverse_E1x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Neoverse_E1x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Neoverse_E1x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_E1x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Neoverse_E1x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Neoverse_E1x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Neoverse_E1x2.bp.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Neoverse_E1x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Neoverse_E1x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Neoverse_E1x2.bp.pl111_clcd_labeller`

Type: `Labeller`.

`FVP_Base_Neoverse_E1x2.bp.pl180_mci`

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Neoverse_E1x2.bp.ps2keyboard`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Neoverse_E1x2.bp.ps2mouse`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Neoverse_E1x2.bp.psram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_E1x2.bp.refcounter`

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

`FVP_Base_Neoverse_E1x2.bp.reset_or`

Or Gate.
Type: `OrGate`.

FVP_Base_Neoverse_E1x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_E1x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_E1x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_E1x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x2.bp.sram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_E1x2.bp.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Neoverse_E1x2.bp.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Neoverse_E1x2.bp.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Neoverse_E1x2.bp.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_Base_Neoverse_E1x2.bp.trusted_key_storage`
Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

`FVP_Base_Neoverse_E1x2.bp.trusted_nv_counter`
Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

`FVP_Base_Neoverse_E1x2.bp.trusted_rng`
Random Number Generator unit.
Type: `RandomNumberGenerator`.

`FVP_Base_Neoverse_E1x2.bp.trusted_watchdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_Base_Neoverse_E1x2.bp.tzc_400`
TrustZone Address Space Controller.
Type: `TZC_400`.

`FVP_Base_Neoverse_E1x2.bp.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_Base_Neoverse_E1x2.bp.vedcc`
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

`FVP_Base_Neoverse_E1x2.bp.virtio_net`
VirtioNet device over MMIO transport.
Type: `VirtioNetMMIO`.

FVP_Base_Neoverse_E1x2.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x2.bp.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_Base_Neoverse_E1x2.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x2.bp.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_Base_Neoverse_E1x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x2.bp.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_Base_Neoverse_E1x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_Base_Neoverse_E1x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Neoverse_E1x2.bp.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_Neoverse_E1x2.cci400

Cache Coherent Interconnect for AXI4 ACE.

Type: CCI400.

FVP_Base_Neoverse_E1x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Neoverse_E1x2.cluster0

ARM Neoverse-E1 Cluster CT model.
Type: Cluster_ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_E1x2.cluster0.cpu0.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x2.cluster0.cpu0.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x2.cluster0.cpu0.l2cache

PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x2.cluster0.cpu0.thread0

ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x2.cluster0.cpu0.thread1

ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x2.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_E1x2.cluster0.cpu1.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x2.cluster0.cpu1.l1icache

PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x2.cluster0.cpu1.l2cache

PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x2.cluster0.cpu1.thread0

ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x2.cluster0.cpu1.thread1

ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x2.cluster0_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x2.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_E1x2.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_E1x2.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_E1x2.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.94 FVP_Base_Neoverse-E1x4

FVP_Base_Neoverse-E1x4 contains the following instances:

FVP_Base_Neoverse-E1x4 instances

FVP_Base_Neoverse_E1x4

Base Platform Compute Subsystem for ARMNeoverseE1x4CT.
Type: FVP_Base_Neoverse_E1x4.

FVP_Base_Neoverse_E1x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_E1x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_E1x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_E1x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_E1x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_E1x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_E1x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_E1x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Neoverse_E1x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Neoverse_E1x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x4.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x4.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x4.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x4.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x4.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x4.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_E1x4.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_E1x4.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_E1x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_E1x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_E1x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_E1x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Neoverse_E1x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Neoverse_E1x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Neoverse_E1x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Neoverse_E1x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Neoverse_E1x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Neoverse_E1x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Neoverse_E1x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Neoverse_E1x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_E1x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Neoverse_E1x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_E1x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_E1x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_E1x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_E1x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_E1x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_E1x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_E1x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_E1x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_E1x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Neoverse_E1x4.bp.pl111_clcd.pl111x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_E1x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Neoverse_E1x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Neoverse_E1x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Neoverse_E1x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Neoverse_E1x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Neoverse_E1x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_E1x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_E1x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_E1x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_E1x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_E1x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x4.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x4.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x4.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_E1x4.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Neoverse_E1x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_E1x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_E1x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_E1x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_E1x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_E1x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_E1x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_E1x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_E1x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_E1x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_E1x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_E1x4.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_E1x4.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_E1x4.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_E1x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_E1x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_E1x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_E1x4.cluster0

ARM Neoverse-E1 Cluster CT model.
Type: Cluster_ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_E1x4.cluster0.cpu0.l1dcache

PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu0.thread0
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu0.thread1
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_E1x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu1.thread0
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu1.thread1
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_E1x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu2.thread0
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu2.thread1
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_E1x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_E1x4.cluster0.cpu3.thread0
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0.cpu3.thread1
ARM Neoverse-E1 CT model.
Type: ARM_Neoverse-E1.

FVP_Base_Neoverse_E1x4.cluster0_labeller
Type: Labeller.

FVP_Base_Neoverse_E1x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_E1x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_E1x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_E1x4.pctl
Base Platforms Power Controller.

Type: Base_PowerController.

7.95 FVP_Base_Neoverse-N1

FVP_Base_Neoverse-N1 contains the following instances:

FVP_Base_Neoverse-N1 instances

FVP_Base_Neoverse_N1
Base Platform Compute Subsystem for ARMNeoverseN1CT.
Type: FVP_Base_Neoverse_N1.

FVP_Base_Neoverse_N1.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_N1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N1.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1.bp.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1.bp.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1.bp.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N1.bp.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.Timer_2_3.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.Timer_2_3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Neoverse_N1.bp.Timer_2_3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Neoverse_N1.bp.ap_refclk`
ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

`FVP_Base_Neoverse_N1.bp.audioout`
SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

`FVP_Base_Neoverse_N1.bp.clock100Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.clock24MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.clock300MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.clock32KHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N1.bp.dmc_phy`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N1.bp.dram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N1.bp.dram_alias_warning`

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Neoverse_N1.bp.dummy_CF`

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Neoverse_N1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_N1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_N1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_N1.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N1.bp.hdlcd0.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a

proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Neoverse_N1.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Neoverse_N1.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Neoverse_N1.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Neoverse_N1.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Neoverse_N1.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Neoverse_N1.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Neoverse_N1.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Neoverse_N1.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Neoverse_N1.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Neoverse_N1.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.pl011_uart2`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Neoverse_N1.bp.pl011_uart2.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.pl011_uart3`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Neoverse_N1.bp.pl011_uart3.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Neoverse_N1.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Neoverse_N1.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Neoverse_N1.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Neoverse_N1.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Neoverse_N1.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Neoverse_N1.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Neoverse_N1.bp.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Neoverse_N1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Neoverse_N1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Neoverse_N1.bp.pl111_clcd_labeller`
Type: `Labeller`.

`FVP_Base_Neoverse_N1.bp.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Neoverse_N1.bp.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Neoverse_N1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Neoverse_N1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Neoverse_N1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Neoverse_N1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_N1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Neoverse_N1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Neoverse_N1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Neoverse_N1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Neoverse_N1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Neoverse_N1.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Neoverse_N1.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Neoverse_N1.bp.trusted_watchdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N1.bp.tzc_400

TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_N1.bp.ve_sysregs

Type: VE_SysRegs.

FVP_Base_Neoverse_N1.bp.vedcc

Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_N1.bp.virtio_net

VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_N1.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Neoverse_N1.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_N1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Neoverse_N1.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_N1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_N1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_N1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_N1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Neoverse_N1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1.cluster0

ARM Neoverse-N1 Cluster CT model.
Type: `Cluster_ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1.cluster0.cpu0

ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1.cluster0.cpu1

ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu2
ARM Neoverse-N1 CT model.
Type: ARM_Neoverse-N1.

FVP_Base_Neoverse_N1.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_N1.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu3
ARM Neoverse-N1 CT model.
Type: ARM_Neoverse-N1.

FVP_Base_Neoverse_N1.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_N1.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1.cluster0_labeller

Type: Labeller.

FVP_Base_Neoverse_N1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_N1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_N1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_N1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.96 FVP_Base_Neoverse-N1x1

FVP_Base_Neoverse-N1x1 contains the following instances:

FVP_Base_Neoverse-N1x1 instances

FVP_Base_Neoverse_N1x1

Base Platform Compute Subsystem for ARMNeoverseN1x1CT.
Type: FVP_Base_Neoverse_N1x1.

FVP_Base_Neoverse_N1x1.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_N1x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N1x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N1x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Neoverse_N1x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Neoverse_N1x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x1.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x1.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x1.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x1.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x1.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x1.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x1.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N1x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N1x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_N1x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_N1x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_N1x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N1x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_N1x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_N1x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_N1x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_N1x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_N1x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_N1x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_N1x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_N1x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_N1x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_N1x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_N1x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_N1x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N1x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_N1x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Neoverse_N1x1.bp.pl111_clcd.pl111x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_N1x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Neoverse_N1x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Neoverse_N1x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Neoverse_N1x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Neoverse_N1x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Neoverse_N1x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Neoverse_N1x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_N1x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N1x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_N1x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Neoverse_N1x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_N1x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_N1x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N1x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_N1x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_N1x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_N1x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_N1x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_N1x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_N1x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_N1x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_N1x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_N1x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_N1x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x1.cluster0

ARM Neoverse-N1 Cluster CT model.
Type: Cluster_ARM_Neoverse-N1.

FVP_Base_Neoverse_N1x1.cluster0.cpu0

ARM Neoverse-N1 CT model.
Type: ARM_Neoverse-N1.

FVP_Base_Neoverse_N1x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_N1x1.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1x1.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1x1.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1x1.cluster0_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_N1x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_N1x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_N1x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.97 FVP_Base_Neoverse-N1x2

FVP_Base_Neoverse-N1x2 contains the following instances:

FVP_Base_Neoverse-N1x2 instances

FVP_Base_Neoverse_N1x2
Base Platform Compute Subsystem for ARMNeoverseN1x2CT.
Type: FVP_Base_Neoverse_N1x2.

FVP_Base_Neoverse_N1x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_N1x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N1x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N1x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N1x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Neoverse_N1x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N1x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N1x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Neoverse_N1x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Neoverse_N1x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N1x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N1x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_N1x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_N1x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_N1x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N1x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_N1x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_N1x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_N1x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_N1x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_N1x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_N1x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x2.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Neoverse_N1x2.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Neoverse_N1x2.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Neoverse_N1x2.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x2.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Neoverse_N1x2.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x2.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Neoverse_N1x2.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Neoverse_N1x2.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Neoverse_N1x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Neoverse_N1x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Neoverse_N1x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Neoverse_N1x2.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Neoverse_N1x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Neoverse_N1x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Neoverse_N1x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Neoverse_N1x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Neoverse_N1x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Neoverse_N1x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_N1x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N1x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_N1x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N1x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Neoverse_N1x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_N1x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_N1x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N1x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_N1x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_N1x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_N1x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_N1x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x2.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_N1x2.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Neoverse_N1x2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_N1x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_N1x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_N1x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_N1x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_N1x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x2.cluster0
ARM Neoverse-N1 Cluster CT model.
Type: `Cluster_ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x2.cluster0.cpu0
ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1x2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x2.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x2.cluster0.cpu1
ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1x2.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x2.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x2.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Neoverse_N1x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Neoverse_N1x2.elfloader
ELF loader component.

Type: ElfLoader.

FVP_Base_Neoverse_N1x2.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_N1x2.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.98 FVP_Base_Neoverse-N1x4

FVP_Base_Neoverse-N1x4 contains the following instances:

FVP_Base_Neoverse-N1x4 instances

FVP_Base_Neoverse_N1x4

Base Platform Compute Subsystem for ARMNeoverseN1x4CT.
Type: FVP_Base_Neoverse_N1x4.

FVP_Base_Neoverse_N1x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_N1x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N1x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N1x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N1x4.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Neoverse_N1x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Neoverse_N1x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x4.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x4.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x4.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x4.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x4.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N1x4.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N1x4.bp.dmc_phy`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N1x4.bp.dram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1x4.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N1x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N1x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x4.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_N1x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_N1x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_N1x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Neoverse_N1x4.bp.hdlcd0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Neoverse_N1x4.bp.hdlcd0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Neoverse_N1x4.bp.hdlcd0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Neoverse_N1x4.bp.hdlcd0_labeller`

Type: `Labeller`.

`FVP_Base_Neoverse_N1x4.bp.hostbridge`

Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Neoverse_N1x4.bp.mmc`

Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Neoverse_N1x4.bp.nontrustedrom`

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Neoverse_N1x4.bp.nontrustedromloader`

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Neoverse_N1x4.bp.pl011_uart0`

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Neoverse_N1x4.bp.pl011_uart0.clk_divider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N1x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_N1x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_N1x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_N1x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_N1x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N1x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_N1x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_N1x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N1x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_N1x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_N1x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_N1x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Neoverse_N1x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).

Type: PL180_MCI.

FVP_Base_Neoverse_N1x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Neoverse_N1x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Neoverse_N1x4.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Neoverse_N1x4.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Neoverse_N1x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N1x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N1x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N1x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_N1x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N1x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_N1x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_N1x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Neoverse_N1x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_N1x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_N1x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N1x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_N1x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_N1x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_N1x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_N1x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_N1x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_N1x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_N1x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_N1x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N1x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Neoverse_N1x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N1x4.cluster0

ARM Neoverse-N1 Cluster CT model.
Type: `Cluster_ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x4.cluster0.cpu0

ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu1
ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu1.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu2
ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu2.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu3
ARM Neoverse-N1 CT model.
Type: `ARM_Neoverse-N1`.

FVP_Base_Neoverse_N1x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N1x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N1x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_N1x4.cluster0_labeller
Type: Labeller.

FVP_Base_Neoverse_N1x4.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_N1x4.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_N1x4.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_N1x4.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.99 FVP_Base_Neoverse-N2

FVP_Base_Neoverse-N2 contains the following instances:

FVP_Base_Neoverse-N2 instances

FVP_Base_Neoverse_N2
Base Platform Compute Subsystem for ARMNeoverseN2CT.
Type: FVP_Base_Neoverse_N2.

FVP_Base_Neoverse_N2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_N2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N2.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Neoverse_N2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_N2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_N2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Neoverse_N2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Neoverse_N2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_N2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_N2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_N2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_N2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_N2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_N2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_N2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_N2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_N2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_N2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_N2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_N2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_N2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_N2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_N2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_N2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread64.

FVP_Base_Neoverse_N2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_N2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_N2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Neoverse_N2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Neoverse_N2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Neoverse_N2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Neoverse_N2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Neoverse_N2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Neoverse_N2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_N2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_N2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Neoverse_N2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_N2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_N2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_N2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_N2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_N2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_N2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_N2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_N2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_N2.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_N2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_N2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_N2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_N2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_N2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2.cluster0

ARM Neoverse-N2 Cluster CT model.
Type: Cluster_ARM_Neoverse-N2.

FVP_Base_Neoverse_N2.cluster0.cpu0

ARM Neoverse-N2 CT model.
Type: ARM_Neoverse-N2.

FVP_Base_Neoverse_N2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N2.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N2.cluster0_labeller
Type: `Labeller`.

FVP_Base_Neoverse_N2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Neoverse_N2.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Neoverse_N2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Neoverse_N2.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.100 FVP_Base_Neoverse-N2x1

FVP_Base_Neoverse-N2x1 contains the following instances:

FVP_Base_Neoverse-N2x1 instances

FVP_Base_Neoverse_N2x1
Base Platform Compute Subsystem for ARMNeoverseN2x1CT.
Type: `FVP_Base_Neoverse_N2x1`.

FVP_Base_Neoverse_N2x1.bp
Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Neoverse_N2x1.bp.Timer_0_1
ARM Dual-Timer Module(SP804).

Type: SP804_Timer.

FVP_Base_Neoverse_N2x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N2x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N2x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_N2x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N2x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_N2x1.bp.ap_refclk

ARM Generic Timer.

Type: MemoryMappedGenericTimer.

FVP_Base_Neoverse_N2x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Neoverse_N2x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N2x1.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N2x1.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N2x1.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N2x1.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N2x1.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Neoverse_N2x1.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Neoverse_N2x1.bp.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N2x1.bp.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N2x1.bp.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_N2x1.bp.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_Base_Neoverse_N2x1.bp.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Neoverse_N2x1.bp.flash1`
Intel Strata Flash J3 LISA+ model.

Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N2x1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2x1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2x1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_N2x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_N2x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_N2x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_N2x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_N2x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_N2x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_N2x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_N2x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_N2x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N2x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_N2x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N2x1.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_Base_Neoverse_N2x1.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_Base_Neoverse_N2x1.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Neoverse_N2x1.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N2x1.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_Base_Neoverse_N2x1.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_N2x1.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_Base_Neoverse_N2x1.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_Base_Neoverse_N2x1.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Neoverse_N2x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Neoverse_N2x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Neoverse_N2x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Neoverse_N2x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Neoverse_N2x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Neoverse_N2x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Neoverse_N2x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Neoverse_N2x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N2x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Neoverse_N2x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Neoverse_N2x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_N2x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_N2x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_N2x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_N2x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N2x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_N2x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2x1.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2x1.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_N2x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Neoverse_N2x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_N2x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_N2x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_N2x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_N2x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_N2x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_N2x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_N2x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_N2x1.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_N2x1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Neoverse_N2x1.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_N2x1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_N2x1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_N2x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_N2x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_N2x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_N2x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_N2x1.cluster0
ARM Neoverse-N2 Cluster CT model.
Type: `Cluster_ARM_Neoverse-N2`.

FVP_Base_Neoverse_N2x1.cluster0.cpu0
ARM Neoverse-N2 CT model.
Type: `ARM_Neoverse-N2`.

FVP_Base_Neoverse_N2x1.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_N2x1.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N2x1.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N2x1.cluster0.cpu0.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_N2x1.cluster0_labeller
Type: `Labeller`.

FVP_Base_Neoverse_N2x1.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_Base_Neoverse_N2x1.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_Base_Neoverse_N2x1.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Neoverse_N2x1.pctl
Base Platforms Power Controller.
Type: `Base_PowerController`.

7.101 FVP_Base_Neoverse-V1

FVP_Base_Neoverse-V1 contains the following instances:

FVP_Base_Neoverse-V1 instances

FVP_Base_Neoverse_V1
Base Platform Compute Subsystem for ARMNeoverseV1CT.

Type: `FVP_Base_Neoverse_V1`.

`FVP_Base_Neoverse_V1.bp`

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

`FVP_Base_Neoverse_V1.bp.Timer_0_1`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Neoverse_V1.bp.Timer_0_1.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1.bp.Timer_0_1.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1.bp.Timer_0_1.counter0`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Neoverse_V1.bp.Timer_0_1.counter1`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_Base_Neoverse_V1.bp.Timer_2_3`

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

`FVP_Base_Neoverse_V1.bp.Timer_2_3.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1.bp.Timer_2_3.clk_div1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1.bp.Timer_2_3.counter0`

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_Base_Neoverse_V1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_V1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Neoverse_V1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Neoverse_V1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1.bp.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1.bp.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1.bp.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_V1.bp.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_V1.bp.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_V1.bp.dram_alias_warning`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Neoverse_V1.bp.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_Base_Neoverse_V1.bp.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_V1.bp.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_V1.bp.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_V1.bp.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Neoverse_V1.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_V1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_V1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_V1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_V1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_V1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_V1.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_V1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_V1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_V1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_V1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_V1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_V1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_V1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_V1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_V1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_Base_Neoverse_V1.bp.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Neoverse_V1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Neoverse_V1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Neoverse_V1.bp.pl111_clcd_labeller`

Type: `Labeller`.

`FVP_Base_Neoverse_V1.bp.pl180_mci`

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_Base_Neoverse_V1.bp.ps2keyboard`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_Base_Neoverse_V1.bp.ps2mouse`

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_Base_Neoverse_V1.bp.psram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_Base_Neoverse_V1.bp.refcounter`

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Neoverse_V1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Neoverse_V1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_V1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_V1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_V1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Neoverse_V1.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Neoverse_V1.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Neoverse_V1.bp.trusted_watchdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Neoverse_V1.bp.tzc_400

TrustZone Address Space Controller.
Type: `TZC_400`.

FVP_Base_Neoverse_V1.bp.ve_sysregs

Type: `VE_SysRegs`.

FVP_Base_Neoverse_V1.bp.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_Base_Neoverse_V1.bp.virtio_net

VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_V1.bp.virtio_net_labeller

Type: Labeller.

FVP_Base_Neoverse_V1.bp.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_V1.bp.virtioblockdevice_labeller

Type: Labeller.

FVP_Base_Neoverse_V1.bp.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_V1.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_V1.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_V1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_V1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_V1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1.cluster0

ARM Neoverse-V1 Cluster CT model.
Type: `Cluster_ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1.cluster0.cpu0

ARM Neoverse-V1 CT model.
Type: `ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_V1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1.cluster0.cpu1

ARM Neoverse-V1 CT model.
Type: `ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_V1.cluster0.cpu1.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1.cluster0.cpu1.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1.cluster0.cpu1.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1.cluster0.cpu2

ARM Neoverse-V1 CT model.

Type: `ARM_Neoverse-V1`.

`FVP_Base_Neoverse_V1.cluster0.cpu2.dtlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Neoverse_V1.cluster0.cpu2.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Neoverse_V1.cluster0.cpu2.l1icache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Neoverse_V1.cluster0.cpu2.l2cache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Neoverse_V1.cluster0.cpu3`
ARM Neoverse-V1 CT model.
Type: `ARM_Neoverse-V1`.

`FVP_Base_Neoverse_V1.cluster0.cpu3.dtlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_Base_Neoverse_V1.cluster0.cpu3.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Neoverse_V1.cluster0.cpu3.l1icache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Neoverse_V1.cluster0.cpu3.l2cache`
PV Cache.
Type: `PVCache`.

`FVP_Base_Neoverse_V1.cluster0_labeller`
Type: `Labeller`.

`FVP_Base_Neoverse_V1.dapmemlogger`
Bus Logger.
Type: `PVBusLogger`.

`FVP_Base_Neoverse_V1.elfloader`
ELF loader component.
Type: `ElfLoader`.

`FVP_Base_Neoverse_V1.gic_distributor`
GIC Interrupt Redistribution Infrastucture component.
Type: `GIC_IRI`.

FVP_Base_Neoverse_V1.pctl

Base Platforms Power Controller.
Type: `Base_PowerController`.

7.102 FVP_Base_Neoverse-V1x1

FVP_Base_Neoverse-V1x1 contains the following instances:

FVP_Base_Neoverse-V1x1 instances

FVP_Base_Neoverse_V1x1

Base Platform Compute Subsystem for ARMNeoverseV1x1CT.
Type: `FVP_Base_Neoverse_V1x1`.

FVP_Base_Neoverse_V1x1.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Neoverse_V1x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Neoverse_V1x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Neoverse_V1x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x1.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Neoverse_V1x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Neoverse_V1x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Neoverse_V1x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Neoverse_V1x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.exclusive_monitor
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_Base_Neoverse_V1x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_V1x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_V1x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Neoverse_V1x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Neoverse_V1x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_Base_Neoverse_V1x1.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_Neoverse_V1x1.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Neoverse_V1x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Neoverse_V1x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Neoverse_V1x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Neoverse_V1x1.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_Neoverse_V1x1.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_Neoverse_V1x1.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_Neoverse_V1x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_V1x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_Base_Neoverse_V1x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_Base_Neoverse_V1x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_Base_Neoverse_V1x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Neoverse_V1x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_Base_Neoverse_V1x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_Base_Neoverse_V1x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_Base_Neoverse_V1x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_Neoverse_V1x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_Neoverse_V1x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_Neoverse_V1x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_Neoverse_V1x1.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_Base_Neoverse_V1x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_Base_Neoverse_V1x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_Base_Neoverse_V1x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Neoverse_V1x1.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Neoverse_V1x1.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Neoverse_V1x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_V1x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Neoverse_V1x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Neoverse_V1x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Neoverse_V1x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Neoverse_V1x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x1.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x1.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x1.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x1.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Neoverse_V1x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_V1x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_V1x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_V1x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_V1x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_V1x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_V1x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_V1x1.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_V1x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_V1x1.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x1.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_V1x1.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_V1x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Neoverse_V1x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x1.cluster0

ARM Neoverse-V1 Cluster CT model.
Type: `Cluster_ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1x1.cluster0.cpu0

ARM Neoverse-V1 CT model.
Type: `ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_V1x1.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1x1.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1x1.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1x1.cluster0_labeller

Type: Labeller.

FVP_Base_Neoverse_V1x1.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_V1x1.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_V1x1.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_V1x1.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.103 FVP_Base_Neoverse-V1x2

FVP_Base_Neoverse-V1x2 contains the following instances:

FVP_Base_Neoverse-V1x2 instances

FVP_Base_Neoverse_V1x2

Base Platform Compute Subsystem for ARMNeoverseV1x2CT.
Type: FVP_Base_Neoverse_V1x2.

FVP_Base_Neoverse_V1x2.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_Neoverse_V1x2.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_V1x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_V1x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_V1x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_Neoverse_V1x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_V1x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_Neoverse_V1x2.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_Neoverse_V1x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_Neoverse_V1x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x2.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x2.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x2.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x2.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x2.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x2.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x2.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x2.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_V1x2.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_Neoverse_V1x2.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_V1x2.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1x2.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1x2.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1x2.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1x2.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_V1x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_V1x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_V1x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_V1x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_V1x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_V1x2.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_Neoverse_V1x2.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_Neoverse_V1x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_Neoverse_V1x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_V1x2.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_V1x2.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_V1x2.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_V1x2.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_V1x2.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_V1x2.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_V1x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_V1x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_Neoverse_V1x2.bp.pl111_clcd.pl111x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_V1x2.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Neoverse_V1x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Neoverse_V1x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Neoverse_V1x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_Neoverse_V1x2.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_Neoverse_V1x2.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_Neoverse_V1x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_Neoverse_V1x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_V1x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_Neoverse_V1x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_V1x2.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_V1x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_V1x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_Neoverse_V1x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_Neoverse_V1x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_Neoverse_V1x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_Neoverse_V1x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_V1x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_V1x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_V1x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_V1x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_V1x2.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x2.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_V1x2.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x2.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_V1x2.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_Neoverse_V1x2.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_V1x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_V1x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_Base_Neoverse_V1x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x2.cluster0

ARM Neoverse-V1 Cluster CT model.
Type: Cluster_ARM_Neoverse-V1.

FVP_Base_Neoverse_V1x2.cluster0.cpu0

ARM Neoverse-V1 CT model.
Type: ARM_Neoverse-V1.

FVP_Base_Neoverse_V1x2.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_V1x2.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x2.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x2.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x2.cluster0.cpu1
ARM Neoverse-V1 CT model.
Type: ARM_Neoverse-V1.

FVP_Base_Neoverse_V1x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_V1x2.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x2.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x2.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x2.cluster0_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x2.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_V1x2.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_V1x2.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_V1x2.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

7.104 FVP_Base_Neoverse-V1x4

FVP_Base_Neoverse-V1x4 contains the following instances:

FVP_Base_Neoverse-V1x4 instances

FVP_Base_Neoverse_V1x4

Base Platform Compute Subsystem for ARMNeoverseV1x4CT.
Type: `FVP_Base_Neoverse_V1x4`.

FVP_Base_Neoverse_V1x4.bp

Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_Base_Neoverse_V1x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Neoverse_V1x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_Base_Neoverse_V1x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_Base_Neoverse_V1x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_Base_Neoverse_V1x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_Base_Neoverse_V1x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Neoverse_V1x4.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_Base_Neoverse_V1x4.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x4.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_Neoverse_V1x4.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_Neoverse_V1x4.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1x4.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_Neoverse_V1x4.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1x4.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_Neoverse_V1x4.bp.generic_watchdog

ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_Neoverse_V1x4.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_Neoverse_V1x4.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_V1x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component

which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_Base_Neoverse_V1x4.bp.hdlcd0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_Base_Neoverse_V1x4.bp.hdlcd0.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_Base_Neoverse_V1x4.bp.hdlcd0_labeller`
Type: `Labeller`.

`FVP_Base_Neoverse_V1x4.bp.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_Base_Neoverse_V1x4.bp.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_Base_Neoverse_V1x4.bp.nontrustedrom`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_Base_Neoverse_V1x4.bp.nontrustedromloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_Base_Neoverse_V1x4.bp.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Neoverse_V1x4.bp.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_Base_Neoverse_V1x4.bp.pl011_uart1`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_Base_Neoverse_V1x4.bp.pl011_uart1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_Neoverse_V1x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_Neoverse_V1x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_Neoverse_V1x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_V1x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_Neoverse_V1x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_Neoverse_V1x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_Neoverse_V1x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_Neoverse_V1x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_Neoverse_V1x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_Neoverse_V1x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_Neoverse_V1x4.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_Neoverse_V1x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_Neoverse_V1x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_Neoverse_V1x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_Base_Neoverse_V1x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_Base_Neoverse_V1x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_Base_Neoverse_V1x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_Neoverse_V1x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_Neoverse_V1x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_Base_Neoverse_V1x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_Base_Neoverse_V1x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_Base_Neoverse_V1x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_Base_Neoverse_V1x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_Base_Neoverse_V1x4.bp.trusted_nv_counter

Trusted Non-Volatile Counter unit.
Type: `NonVolatileCounter`.

FVP_Base_Neoverse_V1x4.bp.trusted_rng

Random Number Generator unit.
Type: `RandomNumberGenerator`.

FVP_Base_Neoverse_V1x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_Neoverse_V1x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_Neoverse_V1x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_Neoverse_V1x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_Neoverse_V1x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_Neoverse_V1x4.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_Neoverse_V1x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_Neoverse_V1x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_Base_Neoverse_V1x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_Neoverse_V1x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_Neoverse_V1x4.bp.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_Neoverse_V1x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_Neoverse_V1x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_Base_Neoverse_V1x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_Neoverse_V1x4.cluster0

ARM Neoverse-V1 Cluster CT model.
Type: `Cluster_ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1x4.cluster0.cpu0

ARM Neoverse-V1 CT model.
Type: `ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_V1x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1x4.cluster0.cpu0.l2cache

PV Cache.
Type: `PVCache`.

FVP_Base_Neoverse_V1x4.cluster0.cpu1

ARM Neoverse-V1 CT model.
Type: `ARM_Neoverse-V1`.

FVP_Base_Neoverse_V1x4.cluster0.cpu1.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_Neoverse_V1x4.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu2
ARM Neoverse-V1 CT model.
Type: ARM_Neoverse-V1.

FVP_Base_Neoverse_V1x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_V1x4.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu3
ARM Neoverse-V1 CT model.
Type: ARM_Neoverse-V1.

FVP_Base_Neoverse_V1x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_Neoverse_V1x4.cluster0.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0.cpu3.l2cache
PV Cache.
Type: PVCache.

FVP_Base_Neoverse_V1x4.cluster0_labeller

Type: Labeller.

FVP_Base_Neoverse_V1x4.dapmemlogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_Neoverse_V1x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_Base_Neoverse_V1x4.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_Neoverse_V1x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

7.105 FVP_Base_RevC-2xAEMvA

FVP_Base_RevC-2xAEMvA contains the following instances:

FVP_Base_RevC-2xAEMvA instances

FVP_Base_RevC_2xAEMvA

Base Platform Compute Subsystem for AEMvACT and AEMvACT.
Type: FVP_Base_RevC_2xAEMvA.

FVP_Base_RevC_2xAEMvA.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_RevC_2xAEMvA.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_RevC_2xAEMvA.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_2xAEMvA.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_2xAEMvA.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_RevC_2xAEMvA.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_2xAEMvA.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_2xAEMvA.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_RevC_2xAEMvA.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_RevC_2xAEMvA.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_RevC_2xAEMvA.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_RevC_2xAEMvA.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_RevC_2xAEMvA.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_RevC_2xAEMvA.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_RevC_2xAEMvA.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_RevC_2xAEMvA.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_RevC_2xAEMvA.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_RevC_2xAEMvA.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_Base_RevC_2xAEMvA.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_RevC_2xAEMvA.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_RevC_2xAEMvA.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_Base_RevC_2xAEMvA.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_RevC_2xAEMvA.bp.hdlcd0_labeller

Type: Labeller.

FVP_Base_RevC_2xAEMvA.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_Base_RevC_2xAEMvA.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_Base_RevC_2xAEMvA.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_RevC_2xAEMvA.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_2xAEMvA.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_RevC_2xAEMvA.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_RevC_2xAEMvA.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_RevC_2xAEMvA.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_RevC_2xAEMvA.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_RevC_2xAEMvA.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_RevC_2xAEMvA.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_RevC_2xAEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_RevC_2xAEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_RevC_2xAEMvA.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_RevC_2xAEMvA.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_RevC_2xAEMvA.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_RevC_2xAEMvA.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_RevC_2xAEMvA.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_RevC_2xAEMvA.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_RevC_2xAEMvA.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_RevC_2xAEMvA.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_RevC_2xAEMvA.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_RevC_2xAEMvA.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_RevC_2xAEMvA.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_RevC_2xAEMvA.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_RevC_2xAEMvA.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_2xAEMvA.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_2xAEMvA.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_2xAEMvA.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_2xAEMvA.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_2xAEMvA.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_2xAEMvA.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_RevC_2xAEMvA.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_RevC_2xAEMvA.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_RevC_2xAEMvA.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_RevC_2xAEMvA.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_RevC_2xAEMvA.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_RevC_2xAEMvA.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_RevC_2xAEMvA.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_RevC_2xAEMvA.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_RevC_2xAEMvA.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_RevC_2xAEMvA.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_RevC_2xAEMvA.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_RevC_2xAEMvA.bp.virtiop9device_labeller

Type: `Labeller`.

FVP_Base_RevC_2xAEMvA.bp.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_Base_RevC_2xAEMvA.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_Base_RevC_2xAEMvA.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_2xAEMvA.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_2xAEMvA.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_Base_RevC_2xAEMvA.cci550

This is a cache coherent interconnect.
Type: `CCI550`.

FVP_Base_RevC_2xAEMvA.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_2xAEMvA.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_2xAEMvA.cluster0

ARM AEMvA Cluster CT model.
Type: `Cluster_ARM_AEM-A_MP`.

FVP_Base_RevC_2xAEMvA.cluster0.cpu0
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_RevC_2xAEMvA.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_2xAEMvA.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster0.cpu1
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_RevC_2xAEMvA.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_2xAEMvA.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster0.cpu2
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_RevC_2xAEMvA.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_2xAEMvA.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster0.cpu3
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_RevC_2xAEMvA.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_RevC_2xAEMvA.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_2xAEMvA.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_2xAEMvA.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_2xAEMvA.cluster0_labeller
Type: `Labeller`.

FVP_Base_RevC_2xAEMvA.cluster1
ARM AEMvA Cluster CT model.
Type: `Cluster_ARM_AEM-A_MP`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu0
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu1
ARM AEM-A MP CT model.
Type: `ARM_AEM-A_MP`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_2xAEMvA.cluster1.cpu1.l1icache
PV Cache.

Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster1.cpu2
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_RevC_2xAEMvA.cluster1.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_2xAEMvA.cluster1.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster1.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster1.cpu3
ARM AEM-A MP CT model.
Type: ARM_AEM-A_MP.

FVP_Base_RevC_2xAEMvA.cluster1.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_2xAEMvA.cluster1.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster1.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster1.l2_cache
PV Cache.
Type: PVCache.

FVP_Base_RevC_2xAEMvA.cluster1_labeller
Type: Labeller.

FVP_Base_RevC_2xAEMvA.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_RevC_2xAEMvA.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_RevC_2xAEMvA.gpu

G76 addon for the Base Platform.
Type: `G76_DB`.

FVP_Base_RevC_2xAEMvA.gpu.gpu

ARM Mali-G76 GPU.
Type: `Mali_G76`.

FVP_Base_RevC_2xAEMvA.pci

PCI addon for the Base Platform.
Type: `BasePlatformPCIRevC`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci

Type: `AHCI_PCI`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.ahci

AHCI controller with attached SATA disks and PCIe interface.
Type: `AHCI_SATA`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.buslogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice

PCI Wrapper for memory mapped components.
Type: `PCIDevice`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice.dmalogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice.incoming_memory_logger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice.lost_mastered_transactions

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice.lost_transactions_to_pcie

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice.msix_pba_logger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice.msix_table_logger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_2xAEMvA.pci.ahci_pci.pcidevice.to_client_memory_logger

Bus Logger.

Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pci_smmuv3

System MMUv3 configured for PCI-E Sub-system.
Type: SMMUv3_FOR_PCIE.

FVP_Base_RevC_2xAEMvA.pci.pci_smmuv3.mmu

SMMUv3 AEM.
Type: SMMUv3AEM.

FVP_Base_RevC_2xAEMvA.pci.pci_smmuv3_msirewriter

Recognise writes to the GITS_TRANSLATER register and rewrite them to go to the GITS_TRANSLATE64R register. The DeviceID is expected to be in the bottom 32 bits of ExtendedID of the transaction. Debug transactions and reads are not rewritten. This component can also, optionally, apply a 16 bit 'label' to the top 16 bits of the MasterID in the same way that the Labeller component does.
Type: MSIRewriter.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0

PCI Wrapper for memory mapped components.
Type: PCIDevice.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0.dmalogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0.incoming_memory_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0.lost_mastered_transactions

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0.lost_transactions_to_pcie

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0.msix_pba_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0.msix_table_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice0.to_client_memory_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1

PCI Wrapper for memory mapped components.
Type: PCIDevice.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1.dmalogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1.incoming_memory_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1.lost_mastered_transactions
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1.lost_transactions_to_pcie
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1.msix_pba_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1.msix_table_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcidevice1.to_client_memory_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pcivirtioblockdevice0
virtio PCI block device.
Type: VirtioPCIBlockDevice.

FVP_Base_RevC_2xAEMvA.pci.pcivirtioblockdevice1
virtio PCI block device.
Type: VirtioPCIBlockDevice.

FVP_Base_RevC_2xAEMvA.pci.pvbus2pci
PVBus to PCI Bridge.
Type: PVBus2PCI.

FVP_Base_RevC_2xAEMvA.pci.pvbus2pci.cfglogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pvbus2pci.devicelogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.pvbus2pci.dmalogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.smmulogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pci.tbu0_pre_smmu_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_2xAEMvA.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

FVP_Base_RevC_2xAEMvA.test_engine
Test Engine for base platform.
Type: TestEngine.

FVP_Base_RevC_2xAEMvA.test_engine.device0
SMMUv3 Test Engine.
Type: SMMUv3TestEngine.

7.106 FVP_Base_RevC-Cortex-A510

FVP_Base_RevC-Cortex-A510 contains the following instances:

FVP_Base_RevC-Cortex-A510 instances

FVP_Base_RevC_Cortex_A510
Base Platform Compute Subsystem for ARMCortexA510CT.
Type: FVP_Base_RevC_Cortex_A510.

FVP_Base_RevC_Cortex_A510.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_Base_RevC_Cortex_A510.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_RevC_Cortex_A510.bp.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_Cortex_A510.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_Cortex_A510.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_Base_RevC_Cortex_A510.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_Cortex_A510.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_Base_RevC_Cortex_A510.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_Base_RevC_Cortex_A510.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_Base_RevC_Cortex_A510.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_Cortex_A510.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_Cortex_A510.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_Cortex_A510.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_Cortex_A510.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_Cortex_A510.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_Cortex_A510.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_Base_RevC_Cortex_A510.bp.dmc

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_RevC_Cortex_A510.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_Base_RevC_Cortex_A510.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_Base_RevC_Cortex_A510.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_RevC_Cortex_A510.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_RevC_Cortex_A510.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_RevC_Cortex_A510.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_RevC_Cortex_A510.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_Base_RevC_Cortex_A510.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_Base_RevC_Cortex_A510.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_Base_RevC_Cortex_A510.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_Base_RevC_Cortex_A510.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_Base_RevC_Cortex_A510.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_Base_RevC_Cortex_A510.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_Base_RevC_Cortex_A510.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_Base_RevC_Cortex_A510.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_Base_RevC_Cortex_A510.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_Base_RevC_Cortex_A510.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_Base_RevC_Cortex_A510.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_Base_RevC_Cortex_A510.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_Base_RevC_Cortex_A510.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_RevC_Cortex_A510.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_Base_RevC_Cortex_A510.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_Base_RevC_Cortex_A510.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_Base_RevC_Cortex_A510.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_Base_RevC_Cortex_A510.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_Base_RevC_Cortex_A510.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_Base_RevC_Cortex_A510.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_Base_RevC_Cortex_A510.bp.pl111_clcd_labeller

Type: Labeller.

FVP_Base_RevC_Cortex_A510.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_Base_RevC_Cortex_A510.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_Base_RevC_Cortex_A510.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_Base_RevC_Cortex_A510.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_Base_RevC_Cortex_A510.bp.reset_or

Or Gate.
Type: OrGate.

FVP_Base_RevC_Cortex_A510.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_Base_RevC_Cortex_A510.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_Base_RevC_Cortex_A510.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_Base_RevC_Cortex_A510.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_RevC_Cortex_A510.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_Base_RevC_Cortex_A510.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_Cortex_A510.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_Cortex_A510.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_Cortex_A510.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_Base_RevC_Cortex_A510.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_Base_RevC_Cortex_A510.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_Base_RevC_Cortex_A510.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_Base_RevC_Cortex_A510.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_Base_RevC_Cortex_A510.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_Base_RevC_Cortex_A510.bp.ve_sysregs
Type: VE_SysRegs.

FVP_Base_RevC_Cortex_A510.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_Base_RevC_Cortex_A510.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_Base_RevC_Cortex_A510.bp.virtio_net_labeller
Type: Labeller.

FVP_Base_RevC_Cortex_A510.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_Base_RevC_Cortex_A510.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_Base_RevC_Cortex_A510.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_Base_RevC_Cortex_A510.bp.virtiop9device_labeller

Type: Labeller.

FVP_Base_RevC_Cortex_A510.bp.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_Base_RevC_Cortex_A510.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_Base_RevC_Cortex_A510.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.bp.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_Base_RevC_Cortex_A510.cci550

This is a cache coherent interconnect.
Type: CCI550.

FVP_Base_RevC_Cortex_A510.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_Base_RevC_Cortex_A510.cluster0

ARM Cortex-A510Cluster CT model.
Type: Cluster_ARM_Cortex-A510.

FVP_Base_RevC_Cortex_A510.cluster0.cpu0

ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_RevC_Cortex_A510.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_Cortex_A510.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu0.l2cache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu1
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_RevC_Cortex_A510.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_Cortex_A510.cluster0.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu1.l2cache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu2
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_RevC_Cortex_A510.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_Cortex_A510.cluster0.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu2.l2cache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu3
ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu3.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu4
ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu4.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu4.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu4.l1icache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu4.l2cache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu5
ARM Cortex-A510CT model.
Type: `ARM_Cortex-A510`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu5.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu5.l1dcache
PV Cache.
Type: `PVCache`.

FVP_Base_RevC_Cortex_A510.cluster0.cpu5.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu5.l2cache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu6
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_RevC_Cortex_A510.cluster0.cpu6.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_Cortex_A510.cluster0.cpu6.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu6.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu6.l2cache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu7
ARM Cortex-A510CT model.
Type: ARM_Cortex-A510.

FVP_Base_RevC_Cortex_A510.cluster0.cpu7.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_Base_RevC_Cortex_A510.cluster0.cpu7.l1dcache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu7.l1icache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0.cpu7.l2cache
PV Cache.
Type: PVCache.

FVP_Base_RevC_Cortex_A510.cluster0_labeller
Type: Labeller.

FVP_Base_RevC_Cortex_A510.dapmemlogger
Bus Logger.

Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.elfloader
ELF loader component.
Type: ElfLoader.

FVP_Base_RevC_Cortex_A510.gic_distributor
GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_Base_RevC_Cortex_A510.pci
PCI addon for the Base Platform.
Type: BasePlatformPCIRevC.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci
Type: AHCI_PCI.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.ahci
AHCI controller with attached SATA disks and PCIe interface.
Type: AHCI_SATA.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.buslogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice
PCI Wrapper for memory mapped components.
Type: PCIDevice.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice.dmalogger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice.incoming_memory_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice.lost_mastered_transactions
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice.lost_transactions_to_pcie
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice.msix_pba_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice.msix_table_logger
Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.ahci_pci.pcidevice.to_client_memory_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pci_smmuv3

System MMUv3 configured for PCI-E Sub-system.
Type: SMMUv3_FOR_PCIE.

FVP_Base_RevC_Cortex_A510.pci.pci_smmuv3.mmu

SMMUv3 AEM.
Type: SMMUv3AEM.

FVP_Base_RevC_Cortex_A510.pci.pci_smmuv3_msirewriter

Recognise writes to the GITS_TRANSLATER register and rewrite them to go to the GITS_TRANSLATE64R register. The DeviceID is expected to be in the bottom 32 bits of ExtendedID of the transaction. Debug transactions and reads are not rewritten. This component can also, optionally, apply a 16 bit 'label' to the top 16 bits of the MasterID in the same way that the Labeller component does.
Type: MSIRewriter.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0

PCI Wrapper for memory mapped components.
Type: PCIDevice.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0.dmalogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0.incoming_memory_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0.lost_mastered_transactions

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0.lost_transactions_to_pcie

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0.msix_pba_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0.msix_table_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pcidevice0.to_client_memory_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1

PCI Wrapper for memory mapped components.
Type: `PCIDevice`.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1.dmalogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1.incoming_memory_logger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1.lost_mastered_transactions

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1.lost_transactions_to_pcie

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1.msix_pba_logger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1.msix_table_logger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pcidevice1.to_client_memory_logger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pcivirtioblockdevice0

virtio PCI block device.
Type: `VirtioPCIBlockDevice`.

FVP_Base_RevC_Cortex_A510.pci.pcivirtioblockdevice1

virtio PCI block device.
Type: `VirtioPCIBlockDevice`.

FVP_Base_RevC_Cortex_A510.pci.pvbus2pci

PVBus to PCI Bridge.
Type: `PVBus2PCI`.

FVP_Base_RevC_Cortex_A510.pci.pvbus2pci.cfglogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pvbus2pci.devicelogger

Bus Logger.
Type: `PVBusLogger`.

FVP_Base_RevC_Cortex_A510.pci.pvbus2pci.dmalogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.smmulogger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pci.tbu0_pre_smmu_logger

Bus Logger.
Type: PVBusLogger.

FVP_Base_RevC_Cortex_A510.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

8 BaseR Platform FVPs

This chapter lists the BaseR Platform FVPs and the instances in them.

For the BaseR Platform memory map, see BaseR Platform memory map in the Fast Models Reference Manual.

8.1 FVP_BaseR_AEMv8R

FVP_BaseR_AEMv8R contains the following instances:

FVP_BaseR_AEMv8R instances

FVP_BaseR_AEMv8R
: Base Platform Compute Subsystem for AEMv8RMPCT.
Type: `FVP_BaseR_AEMv8R`.

FVP_BaseR_AEMv8R.bp
: Peripherals and address map for the Base Platform.
Type: `BasePlatformPeripherals`.

FVP_BaseR_AEMv8R.bp.Timer_0_1
: ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_BaseR_AEMv8R.bp.Timer_0_1.clk_div0
: A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.Timer_0_1.clk_div1
: A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.Timer_0_1.counter0
: Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_AEMv8R.bp.Timer_0_1.counter1
: Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_AEMv8R.bp.Timer_2_3
: ARM Dual-Timer Module(SP804).

Type: SP804_Timer.

FVP_BaseR_AEMv8R.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_AEMv8R.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_AEMv8R.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_AEMv8R.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_AEMv8R.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_BaseR_AEMv8R.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_BaseR_AEMv8R.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_AEMv8R.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_AEMv8R.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_AEMv8R.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_BaseR_AEMv8R.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_BaseR_AEMv8R.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_AEMv8R.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_AEMv8R.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_AEMv8R.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_BaseR_AEMv8R.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_AEMv8R.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_AEMv8R.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_BaseR_AEMv8R.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_BaseR_AEMv8R.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_BaseR_AEMv8R.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_BaseR_AEMv8R.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_BaseR_AEMv8R.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_BaseR_AEMv8R.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_BaseR_AEMv8R.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_BaseR_AEMv8R.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_BaseR_AEMv8R.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_BaseR_AEMv8R.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_BaseR_AEMv8R.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_BaseR_AEMv8R.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_AEMv8R.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_AEMv8R.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_AEMv8R.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_AEMv8R.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_AEMv8R.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_BaseR_AEMv8R.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_BaseR_AEMv8R.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_BaseR_AEMv8R.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_BaseR_AEMv8R.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_BaseR_AEMv8R.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_BaseR_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_BaseR_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_BaseR_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_BaseR_AEMv8R.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_BaseR_AEMv8R.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_BaseR_AEMv8R.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_BaseR_AEMv8R.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_BaseR_AEMv8R.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_BaseR_AEMv8R.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_BaseR_AEMv8R.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_BaseR_AEMv8R.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_BaseR_AEMv8R.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_AEMv8R.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_BaseR_AEMv8R.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_BaseR_AEMv8R.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_BaseR_AEMv8R.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_AEMv8R.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_AEMv8R.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_AEMv8R.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_AEMv8R.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_BaseR_AEMv8R.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_BaseR_AEMv8R.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_BaseR_AEMv8R.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_BaseR_AEMv8R.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_BaseR_AEMv8R.bp.ve_sysregs
Type: VE_SysRegs.

FVP_BaseR_AEMv8R.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_BaseR_AEMv8R.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_BaseR_AEMv8R.bp.virtio_net_labeller
Type: Labeller.

FVP_BaseR_AEMv8R.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_BaseR_AEMv8R.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_BaseR_AEMv8R.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_BaseR_AEMv8R.bp.virtiop9device_labeller
Type: Labeller.

FVP_BaseR_AEMv8R.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_BaseR_AEMv8R.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_BaseR_AEMv8R.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_AEMv8R.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_BaseR_AEMv8R.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_AEMv8R.cluster0

ARMAEMv8-R Cluster CT model.
Type: `Cluster_ARMAEMv8-R_MP`.

FVP_BaseR_AEMv8R.cluster0.cpu0

ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_BaseR_AEMv8R.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_AEMv8R.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.cpu1

ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_BaseR_AEMv8R.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_AEMv8R.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.cpu2
ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_BaseR_AEMv8R.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_AEMv8R.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.cpu3
ARMAEMv8-R MP CT model.
Type: `ARMAEMv8-R_MP`.

FVP_BaseR_AEMv8R.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_AEMv8R.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_BaseR_AEMv8R.cluster0_labeller
Type: `Labeller`.

FVP_BaseR_AEMv8R.dapmemlogger
Bus Logger.

Type: PVBusLogger.

FVP_BaseR_AEMv8R.elfloader

ELF loader component.
Type: ElfLoader.

FVP_BaseR_AEMv8R.flash_ram0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_AEMv8R.gic_distributor

GIC Interrupt Redistribution Infrastucture component.
Type: GIC_IRI.

FVP_BaseR_AEMv8R.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

8.2 FVP_BaseR_Cortex-R52x1

FVP_BaseR_Cortex-R52x1 contains the following instances:

FVP_BaseR_Cortex-R52x1 instances

FVP_BaseR_Cortex_R52x1

Base Platform Compute Subsystem for ARMCortexR52x1CT.
Type: FVP_BaseR_Cortex_R52x1.

FVP_BaseR_Cortex_R52x1.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_BaseR_Cortex_R52x1.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_BaseR_Cortex_R52x1.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_Cortex_R52x1.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_Cortex_R52x1.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_BaseR_Cortex_R52x1.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_Cortex_R52x1.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_Cortex_R52x1.bp.ap_refclk

ARM Generic Timer.
Type: MemoryMappedGenericTimer.

FVP_BaseR_Cortex_R52x1.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_BaseR_Cortex_R52x1.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x1.bp.clock300MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x1.bp.clock32KHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x1.bp.clock35MHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x1.bp.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x1.bp.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x1.bp.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x1.bp.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x1.bp.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.dram_alias_warning
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_BaseR_Cortex_R52x1.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_BaseR_Cortex_R52x1.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_BaseR_Cortex_R52x1.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_Cortex_R52x1.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_Cortex_R52x1.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_BaseR_Cortex_R52x1.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_BaseR_Cortex_R52x1.bp.generic_watchdog
ARM Generic Watchdog.
Type: MemoryMappedGenericWatchdog.

FVP_BaseR_Cortex_R52x1.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_BaseR_Cortex_R52x1.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_BaseR_Cortex_R52x1.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_BaseR_Cortex_R52x1.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_BaseR_Cortex_R52x1.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_BaseR_Cortex_R52x1.bp.hdlcd0_labeller

Type: Labeller.

FVP_BaseR_Cortex_R52x1.bp.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_BaseR_Cortex_R52x1.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_BaseR_Cortex_R52x1.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_Cortex_R52x1.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x1.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_BaseR_Cortex_R52x1.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_BaseR_Cortex_R52x1.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_BaseR_Cortex_R52x1.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_BaseR_Cortex_R52x1.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_BaseR_Cortex_R52x1.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_BaseR_Cortex_R52x1.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_BaseR_Cortex_R52x1.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_BaseR_Cortex_R52x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_BaseR_Cortex_R52x1.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_BaseR_Cortex_R52x1.bp.pl111_clcd_labeller

Type: Labeller.

FVP_BaseR_Cortex_R52x1.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_BaseR_Cortex_R52x1.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_BaseR_Cortex_R52x1.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_BaseR_Cortex_R52x1.bp.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_BaseR_Cortex_R52x1.bp.reset_or

Or Gate.
Type: OrGate.

FVP_BaseR_Cortex_R52x1.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_Cortex_R52x1.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_BaseR_Cortex_R52x1.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_BaseR_Cortex_R52x1.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_BaseR_Cortex_R52x1.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_BaseR_Cortex_R52x1.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x1.bp.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.bp.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x1.bp.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x1.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x1.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x1.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_BaseR_Cortex_R52x1.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_BaseR_Cortex_R52x1.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_BaseR_Cortex_R52x1.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_BaseR_Cortex_R52x1.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_BaseR_Cortex_R52x1.bp.ve_sysregs
Type: VE_SysRegs.

FVP_BaseR_Cortex_R52x1.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_BaseR_Cortex_R52x1.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_BaseR_Cortex_R52x1.bp.virtio_net_labeller
Type: Labeller.

FVP_BaseR_Cortex_R52x1.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_BaseR_Cortex_R52x1.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_BaseR_Cortex_R52x1.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_BaseR_Cortex_R52x1.bp.virtiop9device_labeller

Type: `Labeller`.

FVP_BaseR_Cortex_R52x1.bp.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_BaseR_Cortex_R52x1.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_BaseR_Cortex_R52x1.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x1.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x1.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x1.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_BaseR_Cortex_R52x1.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x1.cluster0

ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x1.cluster0.cpu0

ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x1.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_Cortex_R52x1.cluster0.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_BaseR_Cortex_R52x1.cluster0.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_BaseR_Cortex_R52x1.cluster0.gic_iri
GIC IRI internal to cluster.
Type: gic_iri.

FVP_BaseR_Cortex_R52x1.cluster0_labeller
Type: Labeller.

FVP_BaseR_Cortex_R52x1.dapmemlogger
Bus Logger.
Type: PVBusLogger.

FVP_BaseR_Cortex_R52x1.elfloader
ELF loader component.
Type: ElfLoader.

FVP_BaseR_Cortex_R52x1.flash_ram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x1.pctl
Base Platforms Power Controller.
Type: Base_PowerController.

8.3 FVP_BaseR_Cortex-R52x2

FVP_BaseR_Cortex-R52x2 contains the following instances:

FVP_BaseR_Cortex-R52x2 instances

FVP_BaseR_Cortex_R52x2
Base Platform Compute Subsystem for ARMCortexR52x2CT.
Type: FVP_BaseR_Cortex_R52x2.

FVP_BaseR_Cortex_R52x2.bp
Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_BaseR_Cortex_R52x2.bp.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_BaseR_Cortex_R52x2.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_Cortex_R52x2.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_Cortex_R52x2.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_BaseR_Cortex_R52x2.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_Cortex_R52x2.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_Cortex_R52x2.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_BaseR_Cortex_R52x2.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_BaseR_Cortex_R52x2.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_BaseR_Cortex_R52x2.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x2.bp.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.bp.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_BaseR_Cortex_R52x2.bp.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_BaseR_Cortex_R52x2.bp.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.bp.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.bp.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.bp.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_BaseR_Cortex_R52x2.bp.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_Cortex_R52x2.bp.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_Cortex_R52x2.bp.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_Cortex_R52x2.bp.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_Cortex_R52x2.bp.generic_watchdog

ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_BaseR_Cortex_R52x2.bp.hdlcd0

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_BaseR_Cortex_R52x2.bp.hdlcd0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_BaseR_Cortex_R52x2.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_BaseR_Cortex_R52x2.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_BaseR_Cortex_R52x2.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_BaseR_Cortex_R52x2.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_BaseR_Cortex_R52x2.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_BaseR_Cortex_R52x2.bp.mmc

Generic Multimedia Card.
Type: MMC.

FVP_BaseR_Cortex_R52x2.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_BaseR_Cortex_R52x2.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_BaseR_Cortex_R52x2.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x2.bp.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_BaseR_Cortex_R52x2.bp.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_BaseR_Cortex_R52x2.bp.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_BaseR_Cortex_R52x2.bp.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x2.bp.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_BaseR_Cortex_R52x2.bp.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_BaseR_Cortex_R52x2.bp.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_BaseR_Cortex_R52x2.bp.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_BaseR_Cortex_R52x2.bp.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_BaseR_Cortex_R52x2.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_BaseR_Cortex_R52x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_BaseR_Cortex_R52x2.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_BaseR_Cortex_R52x2.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_BaseR_Cortex_R52x2.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_BaseR_Cortex_R52x2.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_BaseR_Cortex_R52x2.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_BaseR_Cortex_R52x2.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x2.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_BaseR_Cortex_R52x2.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_BaseR_Cortex_R52x2.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x2.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x2.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_BaseR_Cortex_R52x2.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_Cortex_R52x2.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_BaseR_Cortex_R52x2.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_BaseR_Cortex_R52x2.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_BaseR_Cortex_R52x2.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.bp.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x2.bp.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x2.bp.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x2.bp.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_BaseR_Cortex_R52x2.bp.trusted_key_storage
Trusted Root-Key Storage unit.
Type: RootKeyStorage.

FVP_BaseR_Cortex_R52x2.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_BaseR_Cortex_R52x2.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_BaseR_Cortex_R52x2.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_BaseR_Cortex_R52x2.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_BaseR_Cortex_R52x2.bp.ve_sysregs
Type: VE_SysRegs.

FVP_BaseR_Cortex_R52x2.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_BaseR_Cortex_R52x2.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_BaseR_Cortex_R52x2.bp.virtio_net_labeller
Type: Labeller.

FVP_BaseR_Cortex_R52x2.bp.virtioblockdevice

virtio block device.
Type: `VirtioBlockDevice`.

FVP_BaseR_Cortex_R52x2.bp.virtioblockdevice_labeller

Type: `Labeller`.

FVP_BaseR_Cortex_R52x2.bp.virtiop9device

virtio P9 server.
Type: `VirtioP9Device`.

FVP_BaseR_Cortex_R52x2.bp.virtiop9device_labeller

Type: `Labeller`.

FVP_BaseR_Cortex_R52x2.bp.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_BaseR_Cortex_R52x2.bp.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_BaseR_Cortex_R52x2.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x2.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_BaseR_Cortex_R52x2.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x2.cluster0
ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu0
ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu1
ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x2.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x2.cluster0.gic_iri
GIC IRI internal to cluster.
Type: `gic_iri`.

FVP_BaseR_Cortex_R52x2.cluster0_labeller
Type: `Labeller`.

FVP_BaseR_Cortex_R52x2.dapmemlogger
Bus Logger.
Type: `PVBusLogger`.

FVP_BaseR_Cortex_R52x2.elfloader
ELF loader component.
Type: `ElfLoader`.

FVP_BaseR_Cortex_R52x2.flash_ram0
RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_BaseR_Cortex_R52x2.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

8.4 FVP_BaseR_Cortex-R52x4

FVP_BaseR_Cortex-R52x4 contains the following instances:

FVP_BaseR_Cortex-R52x4 instances

FVP_BaseR_Cortex_R52x4

Base Platform Compute Subsystem for ARMCortexR52x4CT.
Type: FVP_BaseR_Cortex_R52x4.

FVP_BaseR_Cortex_R52x4.bp

Peripherals and address map for the Base Platform.
Type: BasePlatformPeripherals.

FVP_BaseR_Cortex_R52x4.bp.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_BaseR_Cortex_R52x4.bp.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x4.bp.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_BaseR_Cortex_R52x4.bp.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_Cortex_R52x4.bp.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_BaseR_Cortex_R52x4.bp.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_BaseR_Cortex_R52x4.bp.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_Cortex_R52x4.bp.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_BaseR_Cortex_R52x4.bp.ap_refclk

ARM Generic Timer.
Type: `MemoryMappedGenericTimer`.

FVP_BaseR_Cortex_R52x4.bp.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_BaseR_Cortex_R52x4.bp.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.clock300MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.clock32KHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.dram_alias_warning

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_BaseR_Cortex_R52x4.bp.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_BaseR_Cortex_R52x4.bp.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.exclusive_monitor
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_BaseR_Cortex_R52x4.bp.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_BaseR_Cortex_R52x4.bp.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_BaseR_Cortex_R52x4.bp.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_Cortex_R52x4.bp.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_Cortex_R52x4.bp.generic_watchdog
ARM Generic Watchdog.
Type: `MemoryMappedGenericWatchdog`.

FVP_BaseR_Cortex_R52x4.bp.hdlcd0
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_BaseR_Cortex_R52x4.bp.hdlcd0.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_BaseR_Cortex_R52x4.bp.hdlcd0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_BaseR_Cortex_R52x4.bp.hdlcd0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_BaseR_Cortex_R52x4.bp.hdlcd0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_BaseR_Cortex_R52x4.bp.hdlcd0_labeller

Type: `Labeller`.

FVP_BaseR_Cortex_R52x4.bp.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_BaseR_Cortex_R52x4.bp.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_BaseR_Cortex_R52x4.bp.nontrustedrom

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_BaseR_Cortex_R52x4.bp.nontrustedromloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_BaseR_Cortex_R52x4.bp.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_BaseR_Cortex_R52x4.bp.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_BaseR_Cortex_R52x4.bp.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_BaseR_Cortex_R52x4.bp.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_BaseR_Cortex_R52x4.bp.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_BaseR_Cortex_R52x4.bp.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_BaseR_Cortex_R52x4.bp.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_BaseR_Cortex_R52x4.bp.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_BaseR_Cortex_R52x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_BaseR_Cortex_R52x4.bp.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_BaseR_Cortex_R52x4.bp.pl111_clcd_labeller

Type: `Labeller`.

FVP_BaseR_Cortex_R52x4.bp.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_BaseR_Cortex_R52x4.bp.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_BaseR_Cortex_R52x4.bp.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_BaseR_Cortex_R52x4.bp.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.refcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_BaseR_Cortex_R52x4.bp.reset_or

Or Gate.
Type: `OrGate`.

FVP_BaseR_Cortex_R52x4.bp.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.secureflash

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_BaseR_Cortex_R52x4.bp.secureflashloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_BaseR_Cortex_R52x4.bp.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_BaseR_Cortex_R52x4.bp.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_BaseR_Cortex_R52x4.bp.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_BaseR_Cortex_R52x4.bp.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.bp.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_Cortex_R52x4.bp.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_Cortex_R52x4.bp.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_Cortex_R52x4.bp.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_BaseR_Cortex_R52x4.bp.trusted_key_storage

Trusted Root-Key Storage unit.
Type: `RootKeyStorage`.

FVP_BaseR_Cortex_R52x4.bp.trusted_nv_counter
Trusted Non-Volatile Counter unit.
Type: NonVolatileCounter.

FVP_BaseR_Cortex_R52x4.bp.trusted_rng
Random Number Generator unit.
Type: RandomNumberGenerator.

FVP_BaseR_Cortex_R52x4.bp.trusted_watchdog
ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_BaseR_Cortex_R52x4.bp.tzc_400
TrustZone Address Space Controller.
Type: TZC_400.

FVP_BaseR_Cortex_R52x4.bp.ve_sysregs
Type: VE_SysRegs.

FVP_BaseR_Cortex_R52x4.bp.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_BaseR_Cortex_R52x4.bp.virtio_net
VirtioNet device over MMIO transport.
Type: VirtioNetMMIO.

FVP_BaseR_Cortex_R52x4.bp.virtio_net_labeller
Type: Labeller.

FVP_BaseR_Cortex_R52x4.bp.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_BaseR_Cortex_R52x4.bp.virtioblockdevice_labeller
Type: Labeller.

FVP_BaseR_Cortex_R52x4.bp.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_BaseR_Cortex_R52x4.bp.virtiop9device_labeller
Type: Labeller.

FVP_BaseR_Cortex_R52x4.bp.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_BaseR_Cortex_R52x4.bp.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_BaseR_Cortex_R52x4.bp.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.bp.vram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_BaseR_Cortex_R52x4.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

FVP_BaseR_Cortex_R52x4.clockdivider0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_BaseR_Cortex_R52x4.cluster0

ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu0

ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu1

ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu2
ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu3
ARM CortexR52 MP CT model.
Type: `ARM_CortexR52`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_BaseR_Cortex_R52x4.cluster0.gic_iri
GIC IRI internal to cluster.
Type: `gic_iri`.

FVP_BaseR_Cortex_R52x4.cluster0_labeller
Type: `Labeller`.

FVP_BaseR_Cortex_R52x4.dapmemlogger
Bus Logger.

Type: PVBusLogger.

FVP_BaseR_Cortex_R52x4.elfloader

ELF loader component.
Type: ElfLoader.

FVP_BaseR_Cortex_R52x4.flash_ram0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_BaseR_Cortex_R52x4.pctl

Base Platforms Power Controller.
Type: Base_PowerController.

9 VE Platform FVPs

This chapter lists the VE Platform FVPs and the instances in them.

For the VE memory maps, see VE memory map for Cortex-A series and VE memory map for Cortex-R series in the Fast Models Reference Manual.

9.1 FVP_VE_Cortex-A15x1

FVP_VE_Cortex-A15x1 contains the following instances:

FVP_VE_Cortex-A15x1 instances

`FVP_VE_Cortex_A15x1`
: Top level component of the Cortex_A15x1 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A15x1`.

`FVP_VE_Cortex_A15x1.cluster`
: ARM Cortex-A15 Cluster CT model.
Type: `Cluster_ARM_Cortex-A15`.

`FVP_VE_Cortex_A15x1.cluster.cpu0`
: ARM Cortex-A15 CT model.
Type: `ARM_Cortex-A15`.

`FVP_VE_Cortex_A15x1.cluster.cpu0.dtlb`
: TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A15x1.cluster.cpu0.itlb`
: TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A15x1.cluster.cpu0.l1dcache`
: PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x1.cluster.cpu0.l1icache`
: PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x1.cluster.l2_cache`
: PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x1.daughterboard`
: Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

FVP_VE_Cortex_A15x1.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x1.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x1.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_VE_Cortex_A15x1.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_VE_Cortex_A15x1.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A15x1.daughterboard.hdlcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A15x1.daughterboard.hdlcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A15x1.daughterboard.hdlcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A15x1.daughterboard.introuter`
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

`FVP_VE_Cortex_A15x1.daughterboard.nonsecure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A15x1.daughterboard.secureDRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x1.daughterboard.secureRO`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A15x1.daughterboard.secureROloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A15x1.daughterboard.secureSRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x1.daughterboard.secure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A15x1.daughterboard.sram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_VE_Cortex_A15x1.globalcounter

Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_VE_Cortex_A15x1.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: `VEMotherBoard`.

FVP_VE_Cortex_A15x1.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A15x1.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A15x1.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A15x1.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A15x1.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x1.motherboard.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x1.motherboard.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A15x1.motherboard.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.clock24MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.clock35MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.clock50Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A15x1.motherboard.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1.motherboard.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x1.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x1.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x1.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x1.motherboard.hostbridge
Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A15x1.motherboard.mmc
Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x1.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A15x1.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A15x1.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x1.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).

Type: PL050_KMI.

FVP_VE_Cortex_A15x1.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A15x1.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A15x1.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A15x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x1.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A15x1.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_VE_Cortex_A15x1.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_VE_Cortex_A15x1.motherboard.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_VE_Cortex_A15x1.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_VE_Cortex_A15x1.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_VE_Cortex_A15x1.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1.motherboard.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1.motherboard.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1.motherboard.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1.motherboard.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1.motherboard.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_VE_Cortex_A15x1.motherboard.virtioblockdevice`
virtio block device.
Type: `VirtioBlockDevice`.

`FVP_VE_Cortex_A15x1.motherboard.virtiop9device`
virtio P9 server.
Type: `VirtioP9Device`.

`FVP_VE_Cortex_A15x1.motherboard.vis`
Display window for VE using Visualisation library.
Type: `VEVisualisation`.

`FVP_VE_Cortex_A15x1.motherboard.vis.recorder`
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

`FVP_VE_Cortex_A15x1.motherboard.vis.recorder.playbackDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1.motherboard.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.2 FVP_VE_Cortex-A15x1-A7x1

FVP_VE_Cortex-A15x1-A7x1 contains the following instances:

FVP_VE_Cortex-A15x1-A7x1 instances

FVP_VE_Cortex_A15x1_A7x1

Top level component of the Cortex A15x1 A7x1 Versatile Express inspired model.
Type: FVP_VE_Cortex_A15x1_A7x1.

FVP_VE_Cortex_A15x1_A7x1.coretile

Dual cluster ARM Cortex-A15x1 and ARM Cortex-A7x1 Core Tile.
Type: ARM_Cortex_A15x1_A7x1_CT.

FVP_VE_Cortex_A15x1_A7x1.coretile.cci400

Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_VE_Cortex_A15x1_A7x1.coretile.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.coretile.clockdivider1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster0

ARM Cortex-A15 Cluster CT model.
Type: Cluster_ARM_Cortex-A15.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster0.cpu0

ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster0.cpu0.dtlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster0.cpu0.itlb

TLB - instruction, data or unified.

Type: `TlbCadi`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster1
ARM Cortex-A7 Cluster CT model.
Type: `Cluster_ARM_Cortex-A7`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster1.cpu0
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster1.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x1_A7x1.coretile.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x1_A7x1.coretile.dualclustersystemconfigurationblock
Dual Cluster System Configuration Block.
Type: `DualClusterSystemConfigurationBlock`.

FVP_VE_Cortex_A15x1_A7x1.coretile.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_VE_Cortex_A15x1_A7x1.coretile.v7_vgic
System VGIC architecture version v7.
Type: `v7_VGIC`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.hdlcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.hdlcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.hdlcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.introuter`
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.nonsecure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.secureDRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.secureRO`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.secureROloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.secureSRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x1_A7x1.daughterboard.secure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1_A7x1.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A15x1_A7x1.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_2_3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.Timer_2_3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.audioout`
SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.clock100Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.clock24MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1_A7x1.motherboard.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A15x1_A7x1.motherboard.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1_A7x1.motherboard.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1_A7x1.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1_A7x1.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x1_A7x1.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x1_A7x1.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x1_A7x1.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x1_A7x1.motherboard.hostbridge
Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A15x1_A7x1.motherboard.mmc
Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).

Type: PL050_KMI.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x1_A7x1.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A15x1_A7x1.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A15x1_A7x1.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A15x1_A7x1.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x1_A7x1.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A15x1_A7x1.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A15x1_A7x1.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A15x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.virtioblockdevice`
virtio block device.
Type: `VirtioBlockDevice`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.virtiop9device`
virtio P9 server.
Type: `VirtioP9Device`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.vis`
Display window for VE using Visualisation library.
Type: `VEVisualisation`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.vis.recorder`
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.vis.recorder.playbackDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x1_A7x1.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x1_A7x1.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.3 FVP_VE_Cortex-A15x2

FVP_VE_Cortex-A15x2 contains the following instances:

FVP_VE_Cortex-A15x2 instances

FVP_VE_Cortex_A15x2
Top level component of the Cortex_A15x2 Versatile Express inspired model.
Type: FVP_VE_Cortex_A15x2.

FVP_VE_Cortex_A15x2.cluster
ARM Cortex-A15 Cluster CT model.
Type: Cluster_ARM_Cortex-A15.

FVP_VE_Cortex_A15x2.cluster.cpu0
ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x2.cluster.cpu0.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x2.cluster.cpu0.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x2.cluster.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x2.cluster.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x2.cluster.cpu1
ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x2.cluster.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x2.cluster.cpu1.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x2.cluster.cpu1.l1dcache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x2.cluster.cpu1.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x2.cluster.l2_cache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x2.daughterboard

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A15x2.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x2.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x2.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x2.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A15x2.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A15x2.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x2.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x2.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A15x2.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x2.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A15x2.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x2.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x2.daughterboard.secureROloader
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x2.daughterboard.secureSRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x2.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A15x2.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A15x2.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A15x2.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x2.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x2.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x2.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x2.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x2.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x2.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A15x2.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A15x2.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x2.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x2.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x2.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A15x2.motherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A15x2.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A15x2.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A15x2.motherboard.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_VE_Cortex_A15x2.motherboard.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x2.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A15x2.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A15x2.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x2.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x2.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A15x2.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A15x2.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_VE_Cortex_A15x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x2.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A15x2.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A15x2.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A15x2.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A15x2.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A15x2.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A15x2.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2.motherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x2.motherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x2.motherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x2.motherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x2.motherboard.ve_sysregs

Type: `VE_SysRegs`.

FVP_VE_Cortex_A15x2.motherboard.virtioblockdevice

virtio block device.
Type: `VirtioBlockDevice`.

FVP_VE_Cortex_A15x2.motherboard.virtiop9device

virtio P9 server.
Type: `VirtioP9Device`.

FVP_VE_Cortex_A15x2.motherboard.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_VE_Cortex_A15x2.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_VE_Cortex_A15x2.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2.motherboard.vram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

9.4 FVP_VE_Cortex-A15x2-A7x2

FVP_VE_Cortex-A15x2-A7x2 contains the following instances:

FVP_VE_Cortex-A15x2-A7x2 instances

`FVP_VE_Cortex_A15x2_A7x2`
Top level component of the Cortex A15x2 A7x2 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A15x2_A7x2`.

`FVP_VE_Cortex_A15x2_A7x2.coretile`
Dual cluster ARM Cortex-A15x2 and ARM Cortex-A7x2 Core Tile.
Type: `ARM_Cortex_A15x2_A7x2_CT`.

`FVP_VE_Cortex_A15x2_A7x2.coretile.cci400`
Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

`FVP_VE_Cortex_A15x2_A7x2.coretile.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.coretile.clockdivider1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0`
ARM Cortex-A15 Cluster CT model.
Type: `Cluster_ARM_Cortex-A15`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu0
ARM Cortex-A15 CT model.
Type: `ARM_Cortex-A15`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu1
ARM Cortex-A15 CT model.
Type: `ARM_Cortex-A15`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu1.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1
ARM Cortex-A7 Cluster CT model.
Type: `Cluster_ARM_Cortex-A7`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu0
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu1
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu1.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x2_A7x2.coretile.dualclustersystemconfigurationblock
Dual Cluster System Configuration Block.
Type: `DualClusterSystemConfigurationBlock`.

FVP_VE_Cortex_A15x2_A7x2.coretile.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_VE_Cortex_A15x2_A7x2.coretile.v7_vgic
System VGIC architecture version v7.
Type: `v7_VGIC`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.hdlcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.hdlcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.hdlcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.introuter`
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.nonsecure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.secureDRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.secureRO`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.secureROloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.secureSRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x2_A7x2.daughterboard.secure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2_A7x2.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A15x2_A7x2.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_2_3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.Timer_2_3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.audioout`
SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.clock100Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.clock24MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2_A7x2.motherboard.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A15x2_A7x2.motherboard.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2_A7x2.motherboard.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2_A7x2.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2_A7x2.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x2_A7x2.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x2_A7x2.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x2_A7x2.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x2_A7x2.motherboard.hostbridge
Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A15x2_A7x2.motherboard.mmc
Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).

Type: PL050_KMI.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x2_A7x2.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A15x2_A7x2.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A15x2_A7x2.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A15x2_A7x2.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x2_A7x2.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A15x2_A7x2.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A15x2_A7x2.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A15x2_A7x2.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x2_A7x2.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.virtioblockdevice`
virtio block device.
Type: `VirtioBlockDevice`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.virtiop9device`
virtio P9 server.
Type: `VirtioP9Device`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.vis`
Display window for VE using Visualisation library.
Type: `VEVisualisation`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.vis.recorder`
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.vis.recorder.playbackDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x2_A7x2.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x2_A7x2.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.5 FVP_VE_Cortex-A15x4

FVP_VE_Cortex-A15x4 contains the following instances:

FVP_VE_Cortex-A15x4 instances

FVP_VE_Cortex_A15x4
Top level component of the Cortex_A15x4 Versatile Express inspired model.
Type: FVP_VE_Cortex_A15x4.

FVP_VE_Cortex_A15x4.cluster
ARM Cortex-A15 Cluster CT model.
Type: Cluster_ARM_Cortex-A15.

FVP_VE_Cortex_A15x4.cluster.cpu0
ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x4.cluster.cpu0.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu0.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.cpu1
ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x4.cluster.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu1.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.cpu2
ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x4.cluster.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu2.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.cpu3
ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x4.cluster.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu3.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4.cluster.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.cluster.l2_cache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4.daughterboard

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

FVP_VE_Cortex_A15x4.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x4.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x4.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x4.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x4.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x4.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x4.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_VE_Cortex_A15x4.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_VE_Cortex_A15x4.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal()

callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A15x4.daughterboard.hdlcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A15x4.daughterboard.hdlcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A15x4.daughterboard.hdlcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A15x4.daughterboard.introuter`
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

`FVP_VE_Cortex_A15x4.daughterboard.nonsecure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A15x4.daughterboard.secureDRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x4.daughterboard.secureRO`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A15x4.daughterboard.secureROloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A15x4.daughterboard.secureSRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x4.daughterboard.secure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A15x4.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x4.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A15x4.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A15x4.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A15x4.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x4.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x4.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x4.motherboard.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x4.motherboard.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A15x4.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A15x4.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A15x4.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x4.motherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_A15x4.motherboard.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x4.motherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x4.motherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A15x4.motherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A15x4.motherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A15x4.motherboard.flashloader0`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A15x4.motherboard.flashloader1`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A15x4.motherboard.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_VE_Cortex_A15x4.motherboard.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_VE_Cortex_A15x4.motherboard.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_VE_Cortex_A15x4.motherboard.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x4.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x4.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x4.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A15x4.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A15x4.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x4.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x4.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A15x4.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A15x4.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A15x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x4.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A15x4.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_VE_Cortex_A15x4.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_VE_Cortex_A15x4.motherboard.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x4.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_VE_Cortex_A15x4.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_VE_Cortex_A15x4.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_VE_Cortex_A15x4.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x4.motherboard.terminal_0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x4.motherboard.terminal_1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x4.motherboard.terminal_2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x4.motherboard.terminal_3`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_VE_Cortex_A15x4.motherboard.ve_sysregs`
Type: `VE_SysRegs`.

`FVP_VE_Cortex_A15x4.motherboard.virtioblockdevice`
virtio block device.
Type: `VirtioBlockDevice`.

`FVP_VE_Cortex_A15x4.motherboard.virtiop9device`
virtio P9 server.
Type: `VirtioP9Device`.

`FVP_VE_Cortex_A15x4.motherboard.vis`
Display window for VE using Visualisation library.
Type: `VEVisualisation`.

`FVP_VE_Cortex_A15x4.motherboard.vis.recorder`
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

`FVP_VE_Cortex_A15x4.motherboard.vis.recorder.playbackDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x4.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.6 FVP_VE_Cortex-A15x4-A7x4

FVP_VE_Cortex-A15x4-A7x4 contains the following instances:

FVP_VE_Cortex-A15x4-A7x4 instances

FVP_VE_Cortex_A15x4_A7x4
Top level component of the Cortex A15x4 A7x4 Versatile Express inspired model.
Type: FVP_VE_Cortex_A15x4_A7x4.

FVP_VE_Cortex_A15x4_A7x4.coretile
Dual cluster ARM Cortex-A15x4 and ARM Cortex-A7x4 Core Tile.
Type: ARM_Cortex_A15x4_A7x4_CT.

FVP_VE_Cortex_A15x4_A7x4.coretile.cci400
Cache Coherent Interconnect for AXI4 ACE.
Type: CCI400.

FVP_VE_Cortex_A15x4_A7x4.coretile.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.coretile.clockdivider1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0
ARM Cortex-A15 Cluster CT model.
Type: Cluster_ARM_Cortex-A15.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu0
ARM Cortex-A15 CT model.
Type: ARM_Cortex-A15.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu0.itlb
TLB - instruction, data or unified.

Type: `TlbCadi`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu0.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu0.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu1`

ARM Cortex-A15 CT model.
Type: `ARM_Cortex-A15`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu1.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu1.itlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu1.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu1.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu2`

ARM Cortex-A15 CT model.
Type: `ARM_Cortex-A15`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu2.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu2.itlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu2.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu2.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu3`

ARM Cortex-A15 CT model.
Type: `ARM_Cortex-A15`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu3.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1
ARM Cortex-A7 Cluster CT model.
Type: `Cluster_ARM_Cortex-A7`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu0
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu1
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu1.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu2
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu2.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu3
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu3.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A15x4_A7x4.coretile.cluster1.l2_cache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A15x4_A7x4.coretile.dualclustersystemconfigurationblock
Dual Cluster System Configuration Block.
Type: DualClusterSystemConfigurationBlock.

FVP_VE_Cortex_A15x4_A7x4.coretile.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A15x4_A7x4.coretile.v7_vgic
System VGIC architecture version v7.
Type: v7_VGIC.

FVP_VE_Cortex_A15x4_A7x4.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.dram_aliased
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.dram_limit_4
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.secure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A15x4_A7x4.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A15x4_A7x4.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x4_A7x4.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A15x4_A7x4.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A15x4_A7x4.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.

Type: SchedulerThreadEvent.

FVP_VE_Cortex_A15x4_A7x4.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).

Type: PL180_MCI.

FVP_VE_Cortex_A15x4_A7x4.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.

Type: PS2Keyboard.

FVP_VE_Cortex_A15x4_A7x4.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.

Type: PS2Mouse.

FVP_VE_Cortex_A15x4_A7x4.motherboard.psram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_VE_Cortex_A15x4_A7x4.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.

Type: SMSC_91C111.

FVP_VE_Cortex_A15x4_A7x4.motherboard.sp805_wdog

ARM Watchdog Module(SP805).

Type: SP805_Watchdog.

FVP_VE_Cortex_A15x4_A7x4.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.

Type: SP810_SysCtrl.

FVP_VE_Cortex_A15x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A15x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.ve_sysregs

Type: `VE_SysRegs`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.virtioblockdevice

virtio block device.
Type: `VirtioBlockDevice`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.virtiop9device

virtio P9 server.
Type: `VirtioP9Device`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_VE_Cortex_A15x4_A7x4.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x4_A7x4.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A15x4_A7x4.motherboard.vram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

9.7 FVP_VE_Cortex-A17x1

FVP_VE_Cortex-A17x1 contains the following instances:

FVP_VE_Cortex-A17x1 instances

`FVP_VE_Cortex_A17x1`
Top level component of the Cortex_A17x1 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A17x1`.

`FVP_VE_Cortex_A17x1.cluster`
ARM Cortex-A17 Cluster CT model.
Type: `Cluster_ARM_Cortex-A17`.

`FVP_VE_Cortex_A17x1.cluster.cpu0`
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

`FVP_VE_Cortex_A17x1.cluster.cpu0.dtlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x1.cluster.cpu0.itlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x1.cluster.cpu0.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x1.cluster.cpu0.l1icache`
PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x1.cluster.l2_cache`
PV Cache.

Type: PVCache.

FVP_VE_Cortex_A17x1.daughterboard

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A17x1.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A17x1.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A17x1.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_A17x1.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A17x1.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A17x1.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A17x1.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_A17x1.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A17x1.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A17x1.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A17x1.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A17x1.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A17x1.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A17x1.gic400
GIC-400 Generic Interrupt Controller.
Type: GIC_400.

FVP_VE_Cortex_A17x1.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A17x1.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A17x1.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x1.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x1.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x1.motherboard.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x1.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A17x1.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A17x1.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A17x1.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1.motherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1.motherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_A17x1.motherboard.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x1.motherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x1.motherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x1.motherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A17x1.motherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A17x1.motherboard.flashloader0`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A17x1.motherboard.flashloader1`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A17x1.motherboard.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_VE_Cortex_A17x1.motherboard.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_VE_Cortex_A17x1.motherboard.pl011_uart0`
ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A17x1.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x1.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x1.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x1.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A17x1.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A17x1.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).

Type: PL050_KMI.

FVP_VE_Cortex_A17x1.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A17x1.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A17x1.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A17x1.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A17x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A17x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A17x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A17x1.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A17x1.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A17x1.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A17x1.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A17x1.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A17x1.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A17x1.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1.motherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1.motherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1.motherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1.motherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1.motherboard.ve_sysregs

Type: `VE_SysRegs`.

FVP_VE_Cortex_A17x1.motherboard.virtioblockdevice

virtio block device.
Type: `VirtioBlockDevice`.

FVP_VE_Cortex_A17x1.motherboard.virtiop9device

virtio P9 server.
Type: `VirtioP9Device`.

FVP_VE_Cortex_A17x1.motherboard.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_VE_Cortex_A17x1.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_VE_Cortex_A17x1.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1.motherboard.vram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

9.8 FVP_VE_Cortex-A17x1-A7x1

FVP_VE_Cortex-A17x1-A7x1 contains the following instances:

FVP_VE_Cortex-A17x1-A7x1 instances

`FVP_VE_Cortex_A17x1_A7x1`
Top level component of the Cortex A17x1 A7x1 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A17x1_A7x1`.

`FVP_VE_Cortex_A17x1_A7x1.coretile`
Dual cluster ARM Cortex-A17x1 and ARM Cortex-A7x1 Core Tile.
Type: `ARM_Cortex_A17x1_A7x1_CT`.

`FVP_VE_Cortex_A17x1_A7x1.coretile.cci400`
Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

`FVP_VE_Cortex_A17x1_A7x1.coretile.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1_A7x1.coretile.clockdivider1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1_A7x1.coretile.cluster0`
ARM Cortex-A17 Cluster CT model.
Type: `Cluster_ARM_Cortex-A17`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster0.cpu0
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster0.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster0.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster0.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster0.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster0.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster1
ARM Cortex-A7 Cluster CT model.
Type: `Cluster_ARM_Cortex-A7`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster1.cpu0
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster1.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster1.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster1.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x1_A7x1.coretile.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x1_A7x1.coretile.dualclustersystemconfigurationblock
Dual Cluster System Configuration Block.
Type: DualClusterSystemConfigurationBlock.

FVP_VE_Cortex_A17x1_A7x1.coretile.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A17x1_A7x1.coretile.v7_vgic
System VGIC architecture version v7.
Type: v7_VGIC.

FVP_VE_Cortex_A17x1_A7x1.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.dram_aliased
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.dram_limit_4
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.dram_limit_8
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.secureROloader
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.secureSRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1_A7x1.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A17x1_A7x1.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.flashloader0`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.flashloader1`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart0`
ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).

Type: PL050_KMI.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl111_clcd.pl111x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A17x1_A7x1.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A17x1_A7x1.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A17x1_A7x1.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A17x1_A7x1.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x1_A7x1.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A17x1_A7x1.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A17x1_A7x1.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A17x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.ve_sysregs

Type: `VE_SysRegs`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.virtioblockdevice

virtio block device.
Type: `VirtioBlockDevice`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.virtiop9device

virtio P9 server.
Type: `VirtioP9Device`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: `VisEventRecorder`.

FVP_VE_Cortex_A17x1_A7x1.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x1_A7x1.motherboard.vram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

9.9 FVP_VE_Cortex-A17x2

FVP_VE_Cortex-A17x2 contains the following instances:

FVP_VE_Cortex-A17x2 instances

`FVP_VE_Cortex_A17x2`
Top level component of the Cortex_A17x2 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A17x2`.

`FVP_VE_Cortex_A17x2.cluster`
ARM Cortex-A17 Cluster CT model.
Type: `Cluster_ARM_Cortex-A17`.

`FVP_VE_Cortex_A17x2.cluster.cpu0`
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

`FVP_VE_Cortex_A17x2.cluster.cpu0.dtlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x2.cluster.cpu0.itlb`
TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x2.cluster.cpu0.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x2.cluster.cpu0.l1icache`
PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x2.cluster.cpu1`
ARM Cortex-A17 CT model.

Type: `ARM_Cortex-A17`.

`FVP_VE_Cortex_A17x2.cluster.cpu1.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x2.cluster.cpu1.itlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x2.cluster.cpu1.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x2.cluster.cpu1.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x2.cluster.l2_cache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x2.daughterboard`

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

`FVP_VE_Cortex_A17x2.daughterboard.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x2.daughterboard.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x2.daughterboard.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x2.daughterboard.dmc_phy`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x2.daughterboard.dram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A17x2.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x2.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x2.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x2.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A17x2.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A17x2.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A17x2.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A17x2.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A17x2.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A17x2.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A17x2.daughterboard.nonsecure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x2.daughterboard.secureDRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x2.daughterboard.secureRO
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A17x2.daughterboard.secureROloader
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A17x2.daughterboard.secureSRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x2.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x2.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x2.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A17x2.gic400
GIC-400 Generic Interrupt Controller.
Type: GIC_400.

FVP_VE_Cortex_A17x2.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A17x2.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A17x2.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x2.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x2.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x2.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x2.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x2.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x2.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A17x2.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A17x2.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x2.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x2.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x2.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A17x2.motherboard.flash1

Intel Strata Flash J3 LISA+ model.

Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A17x2.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A17x2.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A17x2.motherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A17x2.motherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A17x2.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A17x2.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A17x2.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A17x2.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A17x2.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A17x2.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A17x2.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A17x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A17x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A17x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A17x2.motherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_A17x2.motherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_A17x2.motherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_A17x2.motherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x2.motherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_A17x2.motherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_VE_Cortex_A17x2.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A17x2.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x2.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x2.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x2.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x2.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A17x2.motherboard.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A17x2.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A17x2.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A17x2.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A17x2.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x2.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.10 FVP_VE_Cortex-A17x4

FVP_VE_Cortex-A17x4 contains the following instances:

FVP_VE_Cortex-A17x4 instances

FVP_VE_Cortex_A17x4
Top level component of the Cortex_A17x4 Versatile Express inspired model.
Type: FVP_VE_Cortex_A17x4.

FVP_VE_Cortex_A17x4.cluster
ARM Cortex-A17 Cluster CT model.
Type: Cluster_ARM_Cortex-A17.

FVP_VE_Cortex_A17x4.cluster.cpu0
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

FVP_VE_Cortex_A17x4.cluster.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4.cluster.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4.cluster.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4.cluster.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4.cluster.cpu1
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

FVP_VE_Cortex_A17x4.cluster.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4.cluster.cpu1.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4.cluster.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4.cluster.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4.cluster.cpu2
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

FVP_VE_Cortex_A17x4.cluster.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4.cluster.cpu2.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4.cluster.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4.cluster.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4.cluster.cpu3
ARM Cortex-A17 CT model.
Type: ARM_Cortex-A17.

FVP_VE_Cortex_A17x4.cluster.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A17x4.cluster.cpu3.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A17x4.cluster.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4.cluster.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4.cluster.l2_cache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A17x4.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.daughterboard.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.daughterboard.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.daughterboard.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.daughterboard.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.daughterboard.dram_aliased
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4.daughterboard.dram_limit_4
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4.daughterboard.dram_limit_8
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4.daughterboard.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A17x4.daughterboard.hdlcd
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A17x4.daughterboard.hdlcd.timer
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A17x4.daughterboard.hdlcd.timer.timer
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A17x4.daughterboard.hdlcd.timer.timer.thread
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A17x4.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A17x4.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A17x4.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A17x4.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A17x4.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.daughterboard.secure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4.daughterboard.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A17x4.gic400

GIC-400 Generic Interrupt Controller.
Type: GIC_400.

FVP_VE_Cortex_A17x4.globalcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A17x4.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A17x4.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x4.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x4.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x4.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x4.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x4.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x4.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A17x4.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A17x4.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A17x4.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A17x4.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A17x4.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A17x4.motherboard.hostbridge
Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A17x4.motherboard.mmc
Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart1.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart2
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_A17x4.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_VE_Cortex_A17x4.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_VE_Cortex_A17x4.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A17x4.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A17x4.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_VE_Cortex_A17x4.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A17x4.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A17x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A17x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A17x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A17x4.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A17x4.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A17x4.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A17x4.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A17x4.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A17x4.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A17x4.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4.motherboard.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4.motherboard.ve_sysregs
Type: VE_SysRegs.

FVP_VE_Cortex_A17x4.motherboard.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A17x4.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A17x4.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A17x4.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A17x4.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.11 FVP_VE_Cortex-A17x4-A7x4

FVP_VE_Cortex-A17x4-A7x4 contains the following instances:

FVP_VE_Cortex-A17x4-A7x4 instances

FVP_VE_Cortex_A17x4_A7x4
Top level component of the Cortex A17x4 A7x4 Versatile Express inspired model.

Type: `FVP_VE_Cortex_A17x4_A7x4`.

`FVP_VE_Cortex_A17x4_A7x4.coretile`

Dual cluster ARM Cortex-A17x4 and ARM Cortex-A7x4 Core Tile.
Type: `ARM_Cortex_A17x4_A7x4_CT`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cci400`

Cache Coherent Interconnect for AXI4 ACE.
Type: `CCI400`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.clockdivider1`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0`

ARM Cortex-A17 Cluster CT model.
Type: `Cluster_ARM_Cortex-A17`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu0`

ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu0.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu0.itlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu0.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu0.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu1`

ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu1.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu2
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu2.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu3
ARM Cortex-A17 CT model.
Type: `ARM_Cortex-A17`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu3.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster0.l2_cache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1
ARM Cortex-A7 Cluster CT model.
Type: Cluster_ARM_Cortex-A7.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu0
ARM Cortex-A7 CT model.
Type: ARM_Cortex-A7.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu0.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu0.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu1
ARM Cortex-A7 CT model.
Type: ARM_Cortex-A7.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu1.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu1.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu1.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu2
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu2.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu2.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu2.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu2.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu3
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu3.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu3.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu3.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.cpu3.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.cluster1.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A17x4_A7x4.coretile.dualclustersystemconfigurationblock
Dual Cluster System Configuration Block.
Type: `DualClusterSystemConfigurationBlock`.

FVP_VE_Cortex_A17x4_A7x4.coretile.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_VE_Cortex_A17x4_A7x4.coretile.v7_vgic
System VGIC architecture version v7.
Type: v7_VGIC.

FVP_VE_Cortex_A17x4_A7x4.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.dram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.dram_aliased
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.dram_limit_4
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.dram_limit_8
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.exclusive_monitor
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.hdlcd
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4_A7x4.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A17x4_A7x4.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A17x4_A7x4.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.dummy_local_dap_rom`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.flashloader0`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.flashloader1`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A17x4_A7x4.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A17x4_A7x4.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A17x4_A7x4.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A17x4_A7x4.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A17x4_A7x4.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A17x4_A7x4.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A17x4_A7x4.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A17x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4_A7x4.motherboard.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4_A7x4.motherboard.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4_A7x4.motherboard.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A17x4_A7x4.motherboard.ve_sysregs
Type: VE_SysRegs.

FVP_VE_Cortex_A17x4_A7x4.motherboard.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A17x4_A7x4.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A17x4_A7x4.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A17x4_A7x4.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A17x4_A7x4.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A17x4_A7x4.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.12 FVP_VE_Cortex-A5x1

FVP_VE_Cortex-A5x1 contains the following instances:

FVP_VE_Cortex-A5x1 instances

FVP_VE_Cortex_A5x1
Top level component of the Cortex-A5x1 Versatile Express inspired model.
Type: FVP_VE_Cortex_A5x1.

FVP_VE_Cortex_A5x1.cluster
ARM CORTEXA5MP Cluster CT model.
Type: Cluster_ARM_Cortex-A5MP.

FVP_VE_Cortex_A5x1.cluster.cpu0
ARM CORTEXA5MP CT model.
Type: ARM_Cortex-A5MP.

FVP_VE_Cortex_A5x1.cluster.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A5x1.cluster.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A5x1.cluster.cpu0.utlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A5x1.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A5x1.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x1.daughterboard.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A5x1.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x1.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x1.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x1.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x1.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A5x1.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A5x1.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A5x1.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A5x1.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A5x1.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A5x1.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A5x1.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x1.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x1.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A5x1.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A5x1.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x1.daughterboard.secure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x1.daughterboard.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x1.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A5x1.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A5x1.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A5x1.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x1.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x1.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A5x1.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x1.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x1.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A5x1.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x1.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A5x1.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x1.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x1.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x1.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A5x1.motherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A5x1.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A5x1.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A5x1.motherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A5x1.motherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A5x1.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x1.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x1.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x1.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A5x1.motherboard.pl011_uart3`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_VE_Cortex_A5x1.motherboard.pl011_uart3.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A5x1.motherboard.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_VE_Cortex_A5x1.motherboard.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_VE_Cortex_A5x1.motherboard.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_VE_Cortex_A5x1.motherboard.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A5x1.motherboard.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_VE_Cortex_A5x1.motherboard.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A5x1.motherboard.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_VE_Cortex_A5x1.motherboard.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_VE_Cortex_A5x1.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_A5x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A5x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A5x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A5x1.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_VE_Cortex_A5x1.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_VE_Cortex_A5x1.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_VE_Cortex_A5x1.motherboard.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x1.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_VE_Cortex_A5x1.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A5x1.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A5x1.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x1.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x1.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x1.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x1.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A5x1.motherboard.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_VE_Cortex_A5x1.motherboard.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_VE_Cortex_A5x1.motherboard.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_VE_Cortex_A5x1.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_VE_Cortex_A5x1.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A5x1.motherboard.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_VE_Cortex_A5x1.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

9.13 FVP_VE_Cortex-A5x2

FVP_VE_Cortex-A5x2 contains the following instances:

FVP_VE_Cortex-A5x2 instances

FVP_VE_Cortex_A5x2
Top level component of the Cortex-A5x2 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A5x2`.

FVP_VE_Cortex_A5x2.cluster
ARM CORTEXA5MP Cluster CT model.
Type: `Cluster_ARM_Cortex-A5MP`.

FVP_VE_Cortex_A5x2.cluster.cpu0
ARM CORTEXA5MP CT model.
Type: `ARM_Cortex-A5MP`.

FVP_VE_Cortex_A5x2.cluster.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x2.cluster.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x2.cluster.cpu0.utlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A5x2.cluster.cpu1
ARM CORTEXA5MP CT model.
Type: `ARM_Cortex-A5MP`.

FVP_VE_Cortex_A5x2.cluster.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x2.cluster.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x2.cluster.cpu1.utlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A5x2.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

FVP_VE_Cortex_A5x2.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A5x2.daughterboard.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A5x2.daughterboard.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A5x2.daughterboard.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A5x2.daughterboard.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A5x2.daughterboard.dram_aliased`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A5x2.daughterboard.dram_limit_4`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A5x2.daughterboard.dram_limit_8`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A5x2.daughterboard.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_VE_Cortex_A5x2.daughterboard.hdlcd`
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

`FVP_VE_Cortex_A5x2.daughterboard.hdlcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_A5x2.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A5x2.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A5x2.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A5x2.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_A5x2.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A5x2.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x2.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A5x2.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A5x2.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x2.daughterboard.secure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A5x2.daughterboard.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x2.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_VE_Cortex_A5x2.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: `VEMotherBoard`.

FVP_VE_Cortex_A5x2.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A5x2.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x2.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x2.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x2.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x2.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A5x2.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x2.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x2.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x2.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A5x2.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A5x2.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A5x2.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x2.motherboard.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x2.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x2.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A5x2.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A5x2.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A5x2.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A5x2.motherboard.hostbridge
Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A5x2.motherboard.mmc
Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart1.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x2.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A5x2.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A5x2.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A5x2.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A5x2.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A5x2.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A5x2.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A5x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A5x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A5x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A5x2.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A5x2.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A5x2.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A5x2.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x2.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A5x2.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A5x2.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A5x2.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x2.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x2.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x2.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x2.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A5x2.motherboard.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A5x2.motherboard.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A5x2.motherboard.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A5x2.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A5x2.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x2.motherboard.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x2.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.14 FVP_VE_Cortex-A5x4

FVP_VE_Cortex-A5x4 contains the following instances:

FVP_VE_Cortex-A5x4 instances

FVP_VE_Cortex_A5x4
: Top level component of the Cortex-A5x4 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A5x4`.

FVP_VE_Cortex_A5x4.cluster
: ARM CORTEXA5MP Cluster CT model.
Type: `Cluster_ARM_Cortex-A5MP`.

FVP_VE_Cortex_A5x4.cluster.cpu0
: ARM CORTEXA5MP CT model.
Type: `ARM_Cortex-A5MP`.

FVP_VE_Cortex_A5x4.cluster.cpu0.l1dcache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x4.cluster.cpu0.l1icache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x4.cluster.cpu0.utlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A5x4.cluster.cpu1
: ARM CORTEXA5MP CT model.
Type: `ARM_Cortex-A5MP`.

FVP_VE_Cortex_A5x4.cluster.cpu1.l1dcache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x4.cluster.cpu1.l1icache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A5x4.cluster.cpu1.utlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A5x4.cluster.cpu2
: ARM CORTEXA5MP CT model.
Type: `ARM_Cortex-A5MP`.

FVP_VE_Cortex_A5x4.cluster.cpu2.l1dcache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_A5x4.cluster.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_A5x4.cluster.cpu2.utlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A5x4.cluster.cpu3

ARM CORTEXA5MP CT model.
Type: ARM_Cortex-A5MP.

FVP_VE_Cortex_A5x4.cluster.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_A5x4.cluster.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_A5x4.cluster.cpu3.utlb

TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A5x4.daughterboard

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A5x4.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x4.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x4.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x4.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x4.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x4.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A5x4.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A5x4.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A5x4.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A5x4.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A5x4.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A5x4.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A5x4.daughterboard.introuter
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_A5x4.daughterboard.nonsecure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A5x4.daughterboard.secureDRAM
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x4.daughterboard.secureRO
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A5x4.daughterboard.secureROloader
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A5x4.daughterboard.secureSRAM
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x4.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A5x4.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x4.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_VE_Cortex_A5x4.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: `VEMotherBoard`.

FVP_VE_Cortex_A5x4.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A5x4.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x4.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x4.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A5x4.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x4.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A5x4.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A5x4.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A5x4.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x4.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x4.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A5x4.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A5x4.motherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A5x4.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A5x4.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A5x4.motherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A5x4.motherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A5x4.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_VE_Cortex_A5x4.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_VE_Cortex_A5x4.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A5x4.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A5x4.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A5x4.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_VE_Cortex_A5x4.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_VE_Cortex_A5x4.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A5x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A5x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A5x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A5x4.motherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_A5x4.motherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_A5x4.motherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_A5x4.motherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A5x4.motherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_A5x4.motherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_VE_Cortex_A5x4.motherboard.sp810_sysctrl`
Only EB relevant functionalities are fully implemented.

Type: SP810_SysCtrl.

FVP_VE_Cortex_A5x4.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x4.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x4.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x4.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A5x4.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A5x4.motherboard.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A5x4.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A5x4.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A5x4.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A5x4.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A5x4.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A5x4.periph_clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.15 FVP_VE_Cortex-A7x1

FVP_VE_Cortex-A7x1 contains the following instances:

FVP_VE_Cortex-A7x1 instances

FVP_VE_Cortex_A7x1
Top level component of the Cortex_A7x1 Versatile Express inspired model.
Type: FVP_VE_Cortex_A7x1.

FVP_VE_Cortex_A7x1.cluster
ARM Cortex-A7 Cluster CT model.

Type: `Cluster_ARM_Cortex-A7`.

`FVP_VE_Cortex_A7x1.cluster.cpu0`

ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

`FVP_VE_Cortex_A7x1.cluster.cpu0.dtlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A7x1.cluster.cpu0.itlb`

TLB - instruction, data or unified.
Type: `TlbCadi`.

`FVP_VE_Cortex_A7x1.cluster.cpu0.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A7x1.cluster.cpu0.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A7x1.cluster.l2_cache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_A7x1.daughterboard`

Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

`FVP_VE_Cortex_A7x1.daughterboard.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A7x1.daughterboard.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A7x1.daughterboard.dmc`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A7x1.daughterboard.dmc_phy`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x1.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A7x1.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A7x1.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A7x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A7x1.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A7x1.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A7x1.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A7x1.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A7x1.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A7x1.daughterboard.introuter
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_A7x1.daughterboard.nonsecure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A7x1.daughterboard.secureDRAM
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x1.daughterboard.secureRO
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A7x1.daughterboard.secureROloader
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A7x1.daughterboard.secureSRAM
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x1.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A7x1.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x1.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_VE_Cortex_A7x1.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_VE_Cortex_A7x1.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: `VEMotherBoard`.

FVP_VE_Cortex_A7x1.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A7x1.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x1.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x1.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A7x1.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x1.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x1.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A7x1.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A7x1.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x1.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x1.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x1.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A7x1.motherboard.flash1

Intel Strata Flash J3 LISA+ model.

Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A7x1.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A7x1.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A7x1.motherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A7x1.motherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A7x1.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A7x1.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A7x1.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A7x1.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A7x1.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A7x1.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A7x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A7x1.motherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_A7x1.motherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_A7x1.motherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_A7x1.motherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A7x1.motherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_A7x1.motherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_VE_Cortex_A7x1.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A7x1.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x1.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x1.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x1.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x1.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A7x1.motherboard.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A7x1.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A7x1.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A7x1.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A7x1.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x1.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.16 FVP_VE_Cortex-A7x2

FVP_VE_Cortex-A7x2 contains the following instances:

FVP_VE_Cortex-A7x2 instances

FVP_VE_Cortex_A7x2
Top level component of the Cortex_A7x2 Versatile Express inspired model.
Type: FVP_VE_Cortex_A7x2.

FVP_VE_Cortex_A7x2.cluster
ARM Cortex-A7 Cluster CT model.
Type: Cluster_ARM_Cortex-A7.

FVP_VE_Cortex_A7x2.cluster.cpu0
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A7x2.cluster.cpu0.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x2.cluster.cpu0.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x2.cluster.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x2.cluster.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x2.cluster.cpu1
ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A7x2.cluster.cpu1.dtlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x2.cluster.cpu1.itlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x2.cluster.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x2.cluster.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x2.cluster.l2_cache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x2.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

FVP_VE_Cortex_A7x2.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A7x2.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x2.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x2.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x2.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A7x2.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A7x2.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A7x2.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A7x2.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A7x2.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A7x2.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A7x2.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A7x2.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A7x2.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_A7x2.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A7x2.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x2.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A7x2.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A7x2.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x2.daughterboard.secure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A7x2.daughterboard.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x2.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_VE_Cortex_A7x2.globalcounter

Memory Mapped Counter Module for Generic Timers.
Type: MemoryMappedCounterModule.

FVP_VE_Cortex_A7x2.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A7x2.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A7x2.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x2.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x2.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A7x2.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x2.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x2.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A7x2.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_A7x2.motherboard.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x2.motherboard.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x2.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x2.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A7x2.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A7x2.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A7x2.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A7x2.motherboard.hostbridge
Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A7x2.motherboard.mmc
Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_A7x2.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x2.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_VE_Cortex_A7x2.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_VE_Cortex_A7x2.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A7x2.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x2.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A7x2.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x2.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_VE_Cortex_A7x2.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_VE_Cortex_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A7x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A7x2.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_VE_Cortex_A7x2.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.

Type: PS2Keyboard.

FVP_VE_Cortex_A7x2.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A7x2.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A7x2.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A7x2.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A7x2.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A7x2.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x2.motherboard.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x2.motherboard.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x2.motherboard.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x2.motherboard.ve_sysregs
Type: VE_SysRegs.

FVP_VE_Cortex_A7x2.motherboard.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A7x2.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A7x2.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A7x2.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A7x2.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x2.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.17 FVP_VE_Cortex-A7x4

FVP_VE_Cortex-A7x4 contains the following instances:

FVP_VE_Cortex-A7x4 instances

FVP_VE_Cortex_A7x4
: Top level component of the Cortex_A7x3 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A7x4`.

FVP_VE_Cortex_A7x4.cluster
: ARM Cortex-A7 Cluster CT model.
Type: `Cluster_ARM_Cortex-A7`.

FVP_VE_Cortex_A7x4.cluster.cpu0
: ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A7x4.cluster.cpu0.dtlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x4.cluster.cpu0.itlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x4.cluster.cpu0.l1dcache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x4.cluster.cpu0.l1icache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x4.cluster.cpu1
: ARM Cortex-A7 CT model.
Type: `ARM_Cortex-A7`.

FVP_VE_Cortex_A7x4.cluster.cpu1.dtlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x4.cluster.cpu1.itlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A7x4.cluster.cpu1.l1dcache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A7x4.cluster.cpu1.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A7x4.cluster.cpu2
ARM Cortex-A7 CT model.
Type: ARM_Cortex-A7.

FVP_VE_Cortex_A7x4.cluster.cpu2.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A7x4.cluster.cpu2.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A7x4.cluster.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A7x4.cluster.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A7x4.cluster.cpu3
ARM Cortex-A7 CT model.
Type: ARM_Cortex-A7.

FVP_VE_Cortex_A7x4.cluster.cpu3.dtlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A7x4.cluster.cpu3.itlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A7x4.cluster.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A7x4.cluster.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A7x4.cluster.l2_cache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A7x4.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A7x4.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A7x4.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A7x4.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A7x4.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_VE_Cortex_A7x4.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

FVP_VE_Cortex_A7x4.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A7x4.daughterboard.hdlcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A7x4.daughterboard.hdlcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A7x4.daughterboard.hdlcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A7x4.daughterboard.introuter`
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

`FVP_VE_Cortex_A7x4.daughterboard.nonsecure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A7x4.daughterboard.secureDRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A7x4.daughterboard.secureRO`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_A7x4.daughterboard.secureROloader`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_A7x4.daughterboard.secureSRAM`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A7x4.daughterboard.secure_region`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A7x4.daughterboard.sram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_VE_Cortex_A7x4.globalcounter
Memory Mapped Counter Module for Generic Timers.
Type: `MemoryMappedCounterModule`.

FVP_VE_Cortex_A7x4.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: `VEMotherBoard`.

FVP_VE_Cortex_A7x4.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A7x4.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.motherboard.Timer_0_1.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.motherboard.Timer_0_1.counter0
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A7x4.motherboard.Timer_0_1.counter1
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A7x4.motherboard.Timer_2_3
ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A7x4.motherboard.Timer_2_3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.motherboard.Timer_2_3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.Timer_2_3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x4.motherboard.Timer_2_3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_A7x4.motherboard.audioout
SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_A7x4.motherboard.clock100Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.clock24MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.clock35MHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.clock50Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.dummy_CF
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A7x4.motherboard.dummy_local_dap_rom
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.motherboard.dummy_ram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A7x4.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A7x4.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A7x4.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A7x4.motherboard.hostbridge
Host Socket Interface Component.
Type: `HostBridge`.

FVP_VE_Cortex_A7x4.motherboard.mmc
Generic Multimedia Card.
Type: `MMC`.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A7x4.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A7x4.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A7x4.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A7x4.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).

Type: PL050_KMI.

FVP_VE_Cortex_A7x4.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A7x4.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A7x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A7x4.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A7x4.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_VE_Cortex_A7x4.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_VE_Cortex_A7x4.motherboard.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A7x4.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_VE_Cortex_A7x4.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_VE_Cortex_A7x4.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

FVP_VE_Cortex_A7x4.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A7x4.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.terminal_0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x4.motherboard.terminal_1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x4.motherboard.terminal_2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x4.motherboard.terminal_3
Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A7x4.motherboard.ve_sysregs
Type: VE_SysRegs.

FVP_VE_Cortex_A7x4.motherboard.virtioblockdevice
virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A7x4.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A7x4.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A7x4.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A7x4.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A7x4.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

9.18 FVP_VE_Cortex-A9x1

FVP_VE_Cortex-A9x1 contains the following instances:

FVP_VE_Cortex-A9x1 instances

FVP_VE_Cortex_A9x1
Top level component of the Cortex-A9x1 Versatile Express inspired model.
Type: FVP_VE_Cortex_A9x1.

FVP_VE_Cortex_A9x1.cluster
ARM CORTEXA9MP Cluster CT model.
Type: Cluster_ARM_Cortex-A9MP.

FVP_VE_Cortex_A9x1.cluster.cpu0
ARM CORTEXA9MP CT model.
Type: ARM_Cortex-A9MP.

FVP_VE_Cortex_A9x1.cluster.cpu0.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A9x1.cluster.cpu0.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A9x1.cluster.cpu0.utlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A9x1.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A9x1.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x1.daughterboard.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A9x1.daughterboard.dmc

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x1.daughterboard.dmc_phy

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x1.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x1.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x1.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A9x1.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A9x1.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A9x1.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A9x1.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A9x1.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A9x1.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A9x1.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x1.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x1.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A9x1.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x1.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x1.daughterboard.secure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x1.daughterboard.sram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x1.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A9x1.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A9x1.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A9x1.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x1.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x1.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A9x1.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x1.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x1.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A9x1.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x1.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A9x1.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x1.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x1.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x1.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A9x1.motherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A9x1.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x1.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x1.motherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A9x1.motherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A9x1.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x1.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x1.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x1.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A9x1.motherboard.pl011_uart3`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_VE_Cortex_A9x1.motherboard.pl011_uart3.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A9x1.motherboard.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_VE_Cortex_A9x1.motherboard.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_VE_Cortex_A9x1.motherboard.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_VE_Cortex_A9x1.motherboard.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A9x1.motherboard.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_VE_Cortex_A9x1.motherboard.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A9x1.motherboard.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_VE_Cortex_A9x1.motherboard.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_VE_Cortex_A9x1.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_A9x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A9x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A9x1.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A9x1.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

FVP_VE_Cortex_A9x1.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_VE_Cortex_A9x1.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

FVP_VE_Cortex_A9x1.motherboard.psram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x1.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_VE_Cortex_A9x1.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A9x1.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A9x1.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x1.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x1.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x1.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x1.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A9x1.motherboard.virtioblockdevice

virtio block device.

Type: VirtioBlockDevice.

FVP_VE_Cortex_A9x1.motherboard.virtiop9device

virtio P9 server.

Type: VirtioP9Device.

FVP_VE_Cortex_A9x1.motherboard.vis

Display window for VE using Visualisation library.

Type: VEVisualisation.

FVP_VE_Cortex_A9x1.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).

Type: VisEventRecorder.

FVP_VE_Cortex_A9x1.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_A9x1.motherboard.vram

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_VE_Cortex_A9x1.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

9.19 FVP_VE_Cortex-A9x2

FVP_VE_Cortex-A9x2 contains the following instances:

FVP_VE_Cortex-A9x2 instances

FVP_VE_Cortex_A9x2
Top level component of the Cortex-A9x2 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A9x2`.

FVP_VE_Cortex_A9x2.cluster
ARM CORTEXA9MP Cluster CT model.
Type: `Cluster_ARM_Cortex-A9MP`.

FVP_VE_Cortex_A9x2.cluster.cpu0
ARM CORTEXA9MP CT model.
Type: `ARM_Cortex-A9MP`.

FVP_VE_Cortex_A9x2.cluster.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x2.cluster.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x2.cluster.cpu0.utlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A9x2.cluster.cpu1
ARM CORTEXA9MP CT model.
Type: `ARM_Cortex-A9MP`.

FVP_VE_Cortex_A9x2.cluster.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x2.cluster.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x2.cluster.cpu1.utlb
TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A9x2.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: `VEDaughterBoard`.

FVP_VE_Cortex_A9x2.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A9x2.daughterboard.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_A9x2.daughterboard.dmc`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A9x2.daughterboard.dmc_phy`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A9x2.daughterboard.dram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A9x2.daughterboard.dram_aliased`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A9x2.daughterboard.dram_limit_4`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A9x2.daughterboard.dram_limit_8`
Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

`FVP_VE_Cortex_A9x2.daughterboard.exclusive_monitor`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_VE_Cortex_A9x2.daughterboard.hdlcd`
ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: `PL370_HDLCD`.

`FVP_VE_Cortex_A9x2.daughterboard.hdlcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_A9x2.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_A9x2.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_VE_Cortex_A9x2.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_VE_Cortex_A9x2.daughterboard.introuter

Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_A9x2.daughterboard.nonsecure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A9x2.daughterboard.secureDRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x2.daughterboard.secureRO

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A9x2.daughterboard.secureROloader

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_A9x2.daughterboard.secureSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x2.daughterboard.secure_region

Allow TrustZone secure/normal bus signals to be routed separately.
Type: `TZSwitch`.

FVP_VE_Cortex_A9x2.daughterboard.sram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x2.daughterboard.vedcc

Daughterboard Configuration Control (DCC).
Type: `VEDCC`.

FVP_VE_Cortex_A9x2.motherboard

Model inspired by the ARM Versatile Express Motherboard.
Type: `VEMotherBoard`.

FVP_VE_Cortex_A9x2.motherboard.Timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A9x2.motherboard.Timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x2.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x2.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A9x2.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x2.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x2.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A9x2.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x2.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A9x2.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x2.motherboard.dummy_ram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x2.motherboard.dummy_usb
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x2.motherboard.flash0
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A9x2.motherboard.flash1
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A9x2.motherboard.flashloader0
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x2.motherboard.flashloader1
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x2.motherboard.hostbridge
Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A9x2.motherboard.mmc
Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart0
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart1
ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart1.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x2.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_A9x2.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_A9x2.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A9x2.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_A9x2.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_A9x2.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_A9x2.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A9x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A9x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A9x2.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A9x2.motherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_A9x2.motherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_A9x2.motherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_A9x2.motherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x2.motherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_A9x2.motherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_A9x2.motherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_A9x2.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x2.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x2.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x2.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x2.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A9x2.motherboard.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A9x2.motherboard.virtiop9device

virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A9x2.motherboard.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A9x2.motherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A9x2.motherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x2.motherboard.vram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x2.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.20 FVP_VE_Cortex-A9x4

FVP_VE_Cortex-A9x4 contains the following instances:

FVP_VE_Cortex-A9x4 instances

FVP_VE_Cortex_A9x4
: Top level component of the Cortex-A9x4 Versatile Express inspired model.
Type: `FVP_VE_Cortex_A9x4`.

FVP_VE_Cortex_A9x4.cluster
: ARM CORTEXA9MP Cluster CT model.
Type: `Cluster_ARM_Cortex-A9MP`.

FVP_VE_Cortex_A9x4.cluster.cpu0
: ARM CORTEXA9MP CT model.
Type: `ARM_Cortex-A9MP`.

FVP_VE_Cortex_A9x4.cluster.cpu0.l1dcache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x4.cluster.cpu0.l1icache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x4.cluster.cpu0.utlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A9x4.cluster.cpu1
: ARM CORTEXA9MP CT model.
Type: `ARM_Cortex-A9MP`.

FVP_VE_Cortex_A9x4.cluster.cpu1.l1dcache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x4.cluster.cpu1.l1icache
: PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_A9x4.cluster.cpu1.utlb
: TLB - instruction, data or unified.
Type: `TlbCadi`.

FVP_VE_Cortex_A9x4.cluster.cpu2
: ARM CORTEXA9MP CT model.
Type: `ARM_Cortex-A9MP`.

FVP_VE_Cortex_A9x4.cluster.cpu2.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A9x4.cluster.cpu2.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A9x4.cluster.cpu2.utlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A9x4.cluster.cpu3
ARM CORTEXA9MP CT model.
Type: ARM_Cortex-A9MP.

FVP_VE_Cortex_A9x4.cluster.cpu3.l1dcache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A9x4.cluster.cpu3.l1icache
PV Cache.
Type: PVCache.

FVP_VE_Cortex_A9x4.cluster.cpu3.utlb
TLB - instruction, data or unified.
Type: TlbCadi.

FVP_VE_Cortex_A9x4.daughterboard
Daughtercard, inspired by the ARM Versatile Express development platform.
Type: VEDaughterBoard.

FVP_VE_Cortex_A9x4.daughterboard.clockCLCD
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.daughterboard.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.daughterboard.dmc
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x4.daughterboard.dmc_phy
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x4.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x4.daughterboard.dram_aliased

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x4.daughterboard.dram_limit_4

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x4.daughterboard.dram_limit_8

Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x4.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_A9x4.daughterboard.hdlcd

ARM PrimeCell HD Color LCD Controller (Nominal Designation PL370).
Type: PL370_HDLCD.

FVP_VE_Cortex_A9x4.daughterboard.hdlcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_A9x4.daughterboard.hdlcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_A9x4.daughterboard.hdlcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_A9x4.daughterboard.hdlcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_A9x4.daughterboard.introuter
Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_A9x4.daughterboard.nonsecure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x4.daughterboard.secureDRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x4.daughterboard.secureRO
Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_A9x4.daughterboard.secureROloader
A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x4.daughterboard.secureSRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x4.daughterboard.secure_region
Allow TrustZone secure/normal bus signals to be routed separately.
Type: TZSwitch.

FVP_VE_Cortex_A9x4.daughterboard.sram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x4.daughterboard.vedcc
Daughterboard Configuration Control (DCC).
Type: VEDCC.

FVP_VE_Cortex_A9x4.motherboard
Model inspired by the ARM Versatile Express Motherboard.
Type: VEMotherBoard.

FVP_VE_Cortex_A9x4.motherboard.Timer_0_1
ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_A9x4.motherboard.Timer_0_1.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.Timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.Timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x4.motherboard.Timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x4.motherboard.Timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_A9x4.motherboard.Timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.Timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.Timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x4.motherboard.Timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_A9x4.motherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

FVP_VE_Cortex_A9x4.motherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_A9x4.motherboard.dummy_local_dap_rom

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x4.motherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x4.motherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_A9x4.motherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A9x4.motherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_A9x4.motherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x4.motherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_A9x4.motherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_A9x4.motherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_A9x4.motherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_VE_Cortex_A9x4.motherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_VE_Cortex_A9x4.motherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A9x4.motherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_A9x4.motherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_A9x4.motherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_VE_Cortex_A9x4.motherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_VE_Cortex_A9x4.motherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_A9x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_A9x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_A9x4.motherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_A9x4.motherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_A9x4.motherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_A9x4.motherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_A9x4.motherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_A9x4.motherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_A9x4.motherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_VE_Cortex_A9x4.motherboard.sp810_sysctrl`
Only EB relevant functionalities are fully implemented.

Type: SP810_SysCtrl.

FVP_VE_Cortex_A9x4.motherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x4.motherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x4.motherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x4.motherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_A9x4.motherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_A9x4.motherboard.virtioblockdevice

virtio block device.
Type: VirtioBlockDevice.

FVP_VE_Cortex_A9x4.motherboard.virtiop9device
virtio P9 server.
Type: VirtioP9Device.

FVP_VE_Cortex_A9x4.motherboard.vis
Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_A9x4.motherboard.vis.recorder
Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_A9x4.motherboard.vis.recorder.playbackDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.vis.recorder.recordingDivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_A9x4.motherboard.vram
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_A9x4.periph_clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.21 FVP_VE_Cortex-R4

FVP_VE_Cortex-R4 contains the following instances:

FVP_VE_Cortex-R4 instances

FVP_VE_Cortex_R4
Top level component of the Cortex_R4 Versatile Express inspired model.
Type: FVP_VE_Cortex_R4.

FVP_VE_Cortex_R4.daughterboard
Cortex-R4 DaughterBoard for Versatile Express.

Type: `VEDaughterBoardCortex_R4`.

`FVP_VE_Cortex_R4.daughterboard.clockCLCD`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R4.daughterboard.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R4.daughterboard.core`

ARM CORTEXR4 CT model.
Type: `ARM_Cortex-R4`.

`FVP_VE_Cortex_R4.daughterboard.core.cpu0.l1dcache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_R4.daughterboard.core.cpu0.l1icache`

PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_R4.daughterboard.dram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R4.daughterboard.exclusive_monitor`

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_VE_Cortex_R4.daughterboard.pl111_clcd`

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_VE_Cortex_R4.daughterboard.pl111_clcd.pl11x_clcd`

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_VE_Cortex_R4.daughterboard.pl111_clcd.pl11x_clcd.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread.

FVP_VE_Cortex_R4.daughterboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_R4.daughterboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_R4.daughterboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_R4.daughterboard.pl310_l2cc

ARM PrimeCell Level 2 Cache Controller (PL310).
Type: PL310_L2CC.

FVP_VE_Cortex_R4.daughterboard.pl390_gic

Generic Interrupt Controller (PL390).
Type: PL390_GIC.

FVP_VE_Cortex_R4.daughterboard.veinterruptmapper

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_R4.vemotherboard

Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: VEMotherBoardR.

FVP_VE_Cortex_R4.vemotherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_R4.vemotherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_R4.vemotherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R4.vemotherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R4.vemotherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_R4.vemotherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_R4.vemotherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_R4.vemotherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_R4.vemotherboard.hostbridge

Host Socket Interface Component.

Type: HostBridge.

FVP_VE_Cortex_R4.vemotherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart4

ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_R4.vemotherboard.pl011_uart4.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_R4.vemotherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_R4.vemotherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R4.vemotherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R4.vemotherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_R4.vemotherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_R4.vemotherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_R4.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_R4.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_R4.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_R4.vemotherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_R4.vemotherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_R4.vemotherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_R4.vemotherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R4.vemotherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_R4.vemotherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

FVP_VE_Cortex_R4.vemotherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_R4.vemotherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R4.vemotherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R4.vemotherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R4.vemotherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R4.vemotherboard.terminal_4

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R4.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R4.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R4.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R4.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R4.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R4.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R4.vemotherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_R4.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R4.vemotherboard.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_R4.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R4.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R4.vemotherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.22 FVP_VE_Cortex-R5x1

FVP_VE_Cortex-R5x1 contains the following instances:

FVP_VE_Cortex-R5x1 instances

FVP_VE_Cortex_R5x1

Top level component of the Cortex_R5x1 Versatile Express inspired model.
Type: FVP_VE_Cortex_R5x1.

FVP_VE_Cortex_R5x1.daughterboard

Cortex-R5x1 DaughterBoard for Versatile Express.
Type: VEDaughterBoardCortex_R5x1.

FVP_VE_Cortex_R5x1.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x1.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.daughterboard.core

ARM CORTEXR5 Cluster CT model.
Type: `Cluster_ARM_Cortex-R5`.

FVP_VE_Cortex_R5x1.daughterboard.core.cpu0

ARM CORTEXR5 CT model.
Type: `ARM_Cortex-R5`.

FVP_VE_Cortex_R5x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R5x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_VE_Cortex_R5x1.daughterboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_VE_Cortex_R5x1.daughterboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

FVP_VE_Cortex_R5x1.daughterboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_VE_Cortex_R5x1.daughterboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_VE_Cortex_R5x1.daughterboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_VE_Cortex_R5x1.daughterboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_R5x1.daughterboard.pl310_l2cc

ARM PrimeCell Level 2 Cache Controller (PL310).
Type: PL310_L2CC.

FVP_VE_Cortex_R5x1.daughterboard.pl390_gic

Generic Interrupt Controller (PL390).
Type: PL390_GIC.

FVP_VE_Cortex_R5x1.daughterboard.veinterruptmapper

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_R5x1.vemotherboard

Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: VEMotherBoardR.

FVP_VE_Cortex_R5x1.vemotherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_R5x1.vemotherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x1.vemotherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x1.vemotherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x1.vemotherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R5x1.vemotherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R5x1.vemotherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_R5x1.vemotherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R5x1.vemotherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R5x1.vemotherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_R5x1.vemotherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_R5x1.vemotherboard.flashloader0`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_R5x1.vemotherboard.flashloader1`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_R5x1.vemotherboard.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_VE_Cortex_R5x1.vemotherboard.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart4

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R5x1.vemotherboard.pl011_uart4.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_R5x1.vemotherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_R5x1.vemotherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R5x1.vemotherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x1.vemotherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R5x1.vemotherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x1.vemotherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_R5x1.vemotherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_R5x1.vemotherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_R5x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke

wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_R5x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_R5x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_R5x1.vemotherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_R5x1.vemotherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_R5x1.vemotherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_R5x1.vemotherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R5x1.vemotherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_R5x1.vemotherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_VE_Cortex_R5x1.vemotherboard.sp810_sysctrl`
Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

`FVP_VE_Cortex_R5x1.vemotherboard.sp810_sysctrl.clkdiv_clk0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x1.vemotherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x1.vemotherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x1.vemotherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x1.vemotherboard.terminal_4

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R5x1.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R5x1.vemotherboard.ve_sysregs

Type: `VE_SysRegs`.

FVP_VE_Cortex_R5x1.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R5x1.vemotherboard.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_VE_Cortex_R5x1.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R5x1.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x1.vemotherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.23 FVP_VE_Cortex-R5x2

FVP_VE_Cortex-R5x2 contains the following instances:

FVP_VE_Cortex-R5x2 instances

FVP_VE_Cortex_R5x2

Top level component of the Cortex_R5x2 Versatile Express inspired model.
Type: FVP_VE_Cortex_R5x2.

FVP_VE_Cortex_R5x2.daughterboard

Cortex-R5x2 DaughterBoard for Versatile Express.
Type: VEDaughterBoardCortex_R5x2.

FVP_VE_Cortex_R5x2.daughterboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.daughterboard.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.daughterboard.core

ARM CORTEXR5 Cluster CT model.

Type: `Cluster_ARM_Cortex-R5`.

`FVP_VE_Cortex_R5x2.daughterboard.core.cpu0`

ARM CORTEXR5 CT model.
Type: `ARM_Cortex-R5`.

`FVP_VE_Cortex_R5x2.daughterboard.core.cpu1`

ARM CORTEXR5 CT model.
Type: `ARM_Cortex-R5`.

`FVP_VE_Cortex_R5x2.daughterboard.dram`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R5x2.daughterboard.exclusive_monitor`

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_VE_Cortex_R5x2.daughterboard.pl111_clcd`

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_VE_Cortex_R5x2.daughterboard.pl111_clcd.pl11x_clcd`

Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_VE_Cortex_R5x2.daughterboard.pl111_clcd.pl11x_clcd.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_R5x2.daughterboard.pl111_clcd.pl11x_clcd.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_R5x2.daughterboard.pl111_clcd.pl11x_clcd.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_R5x2.daughterboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.

Type: SchedulerThreadEvent.

FVP_VE_Cortex_R5x2.daughterboard.pl310_l2cc

ARM PrimeCell Level 2 Cache Controller (PL310).
Type: PL310_L2CC.

FVP_VE_Cortex_R5x2.daughterboard.pl390_gic

Generic Interrupt Controller (PL390).
Type: PL390_GIC.

FVP_VE_Cortex_R5x2.daughterboard.veinterruptmapper

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_R5x2.vemotherboard

Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: VEMotherBoardR.

FVP_VE_Cortex_R5x2.vemotherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_R5x2.vemotherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x2.vemotherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_R5x2.vemotherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R5x2.vemotherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R5x2.vemotherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_R5x2.vemotherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_R5x2.vemotherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_R5x2.vemotherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_R5x2.vemotherboard.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_VE_Cortex_R5x2.vemotherboard.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart4

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R5x2.vemotherboard.pl011_uart4.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_R5x2.vemotherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_R5x2.vemotherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R5x2.vemotherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R5x2.vemotherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_R5x2.vemotherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_R5x2.vemotherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_R5x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_R5x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_VE_Cortex_R5x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_R5x2.vemotherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_R5x2.vemotherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_R5x2.vemotherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_R5x2.vemotherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R5x2.vemotherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_R5x2.vemotherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_R5x2.vemotherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_R5x2.vemotherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x2.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x2.vemotherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x2.vemotherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x2.vemotherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x2.vemotherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x2.vemotherboard.terminal_4

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R5x2.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R5x2.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x2.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R5x2.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R5x2.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R5x2.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R5x2.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R5x2.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R5x2.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R5x2.vemotherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_R5x2.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R5x2.vemotherboard.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_R5x2.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R5x2.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R5x2.vemotherboard.vis.recorder.recordingDivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

9.24 FVP_VE_Cortex-R7x1

FVP_VE_Cortex-R7x1 contains the following instances:

FVP_VE_Cortex-R7x1 instances

`FVP_VE_Cortex_R7x1`
Top level component of the Cortex_R7x1 Versatile Express inspired model.
Type: `FVP_VE_Cortex_R7x1`.

`FVP_VE_Cortex_R7x1.clockdivider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R7x1.daughterboard`
Cortex_R7x1 DaughterBoard for Versatile Express.
Type: `VEDaughterBoardCortex_R7x1`.

`FVP_VE_Cortex_R7x1.daughterboard.core`
ARM CORTEXR7 Cluster CT model.
Type: `Cluster_ARM_Cortex-R7`.

`FVP_VE_Cortex_R7x1.daughterboard.core.cpu0`
ARM CORTEXR7 CT model.
Type: `ARM_Cortex-R7`.

`FVP_VE_Cortex_R7x1.daughterboard.core.cpu0.l1dcache`
PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_R7x1.daughterboard.core.cpu0.l1icache`
PV Cache.
Type: `PVCache`.

`FVP_VE_Cortex_R7x1.daughterboard.dram`
RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_VE_Cortex_R7x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_R7x1.daughterboard.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x1.daughterboard.pl310_l2cc

ARM PrimeCell Level 2 Cache Controller (PL310).
Type: PL310_L2CC.

FVP_VE_Cortex_R7x1.daughterboard.veinterruptmapper

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_R7x1.vemotherboard

Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: VEMotherBoardR.

FVP_VE_Cortex_R7x1.vemotherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_R7x1.vemotherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x1.vemotherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x1.vemotherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x1.vemotherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_VE_Cortex_R7x1.vemotherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R7x1.vemotherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R7x1.vemotherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_R7x1.vemotherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_R7x1.vemotherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_R7x1.vemotherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

FVP_VE_Cortex_R7x1.vemotherboard.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_VE_Cortex_R7x1.vemotherboard.mmc

Generic Multimedia Card.
Type: `MMC`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart4

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R7x1.vemotherboard.pl011_uart4.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_R7x1.vemotherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_R7x1.vemotherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R7x1.vemotherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x1.vemotherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R7x1.vemotherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x1.vemotherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_R7x1.vemotherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_R7x1.vemotherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_R7x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke

wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_R7x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_R7x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_R7x1.vemotherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_R7x1.vemotherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_R7x1.vemotherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_R7x1.vemotherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R7x1.vemotherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_R7x1.vemotherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_VE_Cortex_R7x1.vemotherboard.sp810_sysctrl`
Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

`FVP_VE_Cortex_R7x1.vemotherboard.sp810_sysctrl.clkdiv_clk0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R7x1.vemotherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R7x1.vemotherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R7x1.vemotherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R7x1.vemotherboard.terminal_4

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R7x1.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R7x1.vemotherboard.ve_sysregs

Type: `VE_SysRegs`.

FVP_VE_Cortex_R7x1.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R7x1.vemotherboard.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_VE_Cortex_R7x1.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R7x1.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x1.vemotherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.25 FVP_VE_Cortex-R7x2

FVP_VE_Cortex-R7x2 contains the following instances:

FVP_VE_Cortex-R7x2 instances

FVP_VE_Cortex_R7x2

Top level component of the Cortex_R7x2 Versatile Express inspired model.
Type: FVP_VE_Cortex_R7x2.

FVP_VE_Cortex_R7x2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.daughterboard

Cortex_R7x2 DaughterBoard for Versatile Express.
Type: VEDaughterBoardCortex_R7x2.

FVP_VE_Cortex_R7x2.daughterboard.core

ARM CORTEXR7 Cluster CT model.
Type: Cluster_ARM_Cortex-R7.

FVP_VE_Cortex_R7x2.daughterboard.core.cpu0

ARM CORTEXR7 CT model.
Type: ARM_Cortex-R7.

FVP_VE_Cortex_R7x2.daughterboard.core.cpu0.l1dcache

PV Cache.

Type: PVCache.

FVP_VE_Cortex_R7x2.daughterboard.core.cpu0.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R7x2.daughterboard.core.cpu1

ARM CORTEXR7 CT model.
Type: ARM_Cortex-R7.

FVP_VE_Cortex_R7x2.daughterboard.core.cpu1.l1dcache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R7x2.daughterboard.core.cpu1.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R7x2.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R7x2.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_R7x2.daughterboard.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.daughterboard.pl310_l2cc

ARM PrimeCell Level 2 Cache Controller (PL310).
Type: PL310_L2CC.

FVP_VE_Cortex_R7x2.daughterboard.veinterruptmapper

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_R7x2.vemotherboard

Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: VEMotherBoardR.

FVP_VE_Cortex_R7x2.vemotherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_R7x2.vemotherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R7x2.vemotherboard.clock24MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R7x2.vemotherboard.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R7x2.vemotherboard.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R7x2.vemotherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R7x2.vemotherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_R7x2.vemotherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R7x2.vemotherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R7x2.vemotherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_R7x2.vemotherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

FVP_VE_Cortex_R7x2.vemotherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_R7x2.vemotherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_R7x2.vemotherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_R7x2.vemotherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart4

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R7x2.vemotherboard.pl011_uart4.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x2.vemotherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

FVP_VE_Cortex_R7x2.vemotherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

FVP_VE_Cortex_R7x2.vemotherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_R7x2.vemotherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x2.vemotherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

FVP_VE_Cortex_R7x2.vemotherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R7x2.vemotherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

FVP_VE_Cortex_R7x2.vemotherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_R7x2.vemotherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_R7x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_R7x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_R7x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_R7x2.vemotherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_R7x2.vemotherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_R7x2.vemotherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_R7x2.vemotherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R7x2.vemotherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_R7x2.vemotherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_R7x2.vemotherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_R7x2.vemotherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.terminal_0

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R7x2.vemotherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R7x2.vemotherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R7x2.vemotherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R7x2.vemotherboard.terminal_4

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R7x2.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R7x2.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R7x2.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R7x2.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R7x2.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R7x2.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R7x2.vemotherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_R7x2.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R7x2.vemotherboard.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_R7x2.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R7x2.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R7x2.vemotherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.26 FVP_VE_Cortex-R8x1

FVP_VE_Cortex-R8x1 contains the following instances:

FVP_VE_Cortex-R8x1 instances

FVP_VE_Cortex_R8x1

Top level component of the Cortex_R8x1 Versatile Express inspired model.
Type: FVP_VE_Cortex_R8x1.

FVP_VE_Cortex_R8x1.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x1.daughterboard

Cortex_R8x1 DaughterBoard for Versatile Express.
Type: `VEDaughterBoardCortex_R8x1`.

FVP_VE_Cortex_R8x1.daughterboard.core

ARM CORTEXR8 Cluster CT model.
Type: `Cluster_ARM_Cortex-R8`.

FVP_VE_Cortex_R8x1.daughterboard.core.cpu0

ARM CORTEXR8 CT model.
Type: `ARM_Cortex-R8`.

FVP_VE_Cortex_R8x1.daughterboard.core.cpu0.l1dcache

PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_R8x1.daughterboard.core.cpu0.l1icache

PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_R8x1.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R8x1.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_VE_Cortex_R8x1.daughterboard.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x1.daughterboard.pl310_l2cc

ARM PrimeCell Level 2 Cache Controller (PL310).
Type: `PL310_L2CC`.

FVP_VE_Cortex_R8x1.daughterboard.veinterruptmapper

Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

FVP_VE_Cortex_R8x1.vemotherboard

Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: `VEMotherBoardR`.

FVP_VE_Cortex_R8x1.vemotherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_R8x1.vemotherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.clockCLCD

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.dummy_CF

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_VE_Cortex_R8x1.vemotherboard.dummy_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R8x1.vemotherboard.dummy_usb

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R8x1.vemotherboard.flash0

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_R8x1.vemotherboard.flash1

Intel Strata Flash J3 LISA+ model.
Type: IntelStrataFlashJ3.

FVP_VE_Cortex_R8x1.vemotherboard.flashloader0

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_R8x1.vemotherboard.flashloader1

A device that can preload a gzipped image into flash at startup.
Type: FlashLoader.

FVP_VE_Cortex_R8x1.vemotherboard.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_VE_Cortex_R8x1.vemotherboard.mmc

Generic Multimedia Card.
Type: MMC.

FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart0

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart3`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart3.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart4`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl011_uart4.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl031_rtc`
ARM PrimeCell Real Time Clock(PL031).
Type: `PL031_RTC`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl041_aaci`
ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: `PL041_AACI`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl050_kmi0`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl050_kmi0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl050_kmi1`
ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: `PL050_KMI`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl050_kmi1.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl111_clcd`
ARM PrimeCell Color LCD Controller(PL111).
Type: `PL111_CLCD`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl111_clcd.pl11x_clcd`
Internal component used by PL110 and PL111 CLCD controllers.
Type: `PL11x_CLCD`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl111_clcd.pl11x_clcd.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_R8x1.vemotherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_R8x1.vemotherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

FVP_VE_Cortex_R8x1.vemotherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_R8x1.vemotherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R8x1.vemotherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_R8x1.vemotherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_R8x1.vemotherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_R8x1.vemotherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.terminal_0

Telnet terminal interface.

Type: TelnetTerminal.

FVP_VE_Cortex_R8x1.vemotherboard.terminal_1

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R8x1.vemotherboard.terminal_2

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R8x1.vemotherboard.terminal_3

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R8x1.vemotherboard.terminal_4

Telnet terminal interface.
Type: TelnetTerminal.

FVP_VE_Cortex_R8x1.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R8x1.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x1.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x1.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R8x1.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x1.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x1.vemotherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_R8x1.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R8x1.vemotherboard.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_R8x1.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R8x1.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x1.vemotherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.27 FVP_VE_Cortex-R8x2

FVP_VE_Cortex-R8x2 contains the following instances:

FVP_VE_Cortex-R8x2 instances

FVP_VE_Cortex_R8x2
Top level component of the Cortex_R8x2 Versatile Express inspired model.
Type: `FVP_VE_Cortex_R8x2`.

FVP_VE_Cortex_R8x2.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.daughterboard
Cortex_R8x2 DaughterBoard for Versatile Express.
Type: `VEDaughterBoardCortex_R8x2`.

FVP_VE_Cortex_R8x2.daughterboard.core
ARM CORTEXR8 Cluster CT model.
Type: `Cluster_ARM_Cortex-R8`.

FVP_VE_Cortex_R8x2.daughterboard.core.cpu0
ARM CORTEXR8 CT model.
Type: `ARM_Cortex-R8`.

FVP_VE_Cortex_R8x2.daughterboard.core.cpu0.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_R8x2.daughterboard.core.cpu0.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_R8x2.daughterboard.core.cpu1
ARM CORTEXR8 CT model.
Type: `ARM_Cortex-R8`.

FVP_VE_Cortex_R8x2.daughterboard.core.cpu1.l1dcache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_R8x2.daughterboard.core.cpu1.l1icache
PV Cache.
Type: `PVCache`.

FVP_VE_Cortex_R8x2.daughterboard.dram
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R8x2.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_R8x2.daughterboard.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x2.daughterboard.pl310_l2cc

ARM PrimeCell Level 2 Cache Controller (PL310).
Type: PL310_L2CC.

FVP_VE_Cortex_R8x2.daughterboard.veinterruptmapper

Interrupt Mapping peripheral (non-cascaded).
Type: VEInterruptMapper.

FVP_VE_Cortex_R8x2.vemotherboard

Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: VEMotherBoardR.

FVP_VE_Cortex_R8x2.vemotherboard.audioout

SDL based Audio Output for PL041_AACI.
Type: AudioOut_SDL.

FVP_VE_Cortex_R8x2.vemotherboard.clock100Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x2.vemotherboard.clock24MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x2.vemotherboard.clock35MHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x2.vemotherboard.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x2.vemotherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x2.vemotherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_R8x2.vemotherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R8x2.vemotherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R8x2.vemotherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_R8x2.vemotherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_R8x2.vemotherboard.flashloader0`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_R8x2.vemotherboard.flashloader1`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_R8x2.vemotherboard.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_VE_Cortex_R8x2.vemotherboard.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart1

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart4

ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

FVP_VE_Cortex_R8x2.vemotherboard.pl011_uart4.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_R8x2.vemotherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_R8x2.vemotherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R8x2.vemotherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x2.vemotherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R8x2.vemotherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x2.vemotherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_R8x2.vemotherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_R8x2.vemotherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_R8x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke

wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_VE_Cortex_R8x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_VE_Cortex_R8x2.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_VE_Cortex_R8x2.vemotherboard.pl180_mci`
ARM PrimeCell Multimedia Card Interface (PL180).
Type: `PL180_MCI`.

`FVP_VE_Cortex_R8x2.vemotherboard.ps2keyboard`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Keyboard`.

`FVP_VE_Cortex_R8x2.vemotherboard.ps2mouse`
Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: `PS2Mouse`.

`FVP_VE_Cortex_R8x2.vemotherboard.psram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R8x2.vemotherboard.smsc_91c111`
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_VE_Cortex_R8x2.vemotherboard.sp805_wdog`
ARM Watchdog Module(SP805).
Type: `SP805_Watchdog`.

`FVP_VE_Cortex_R8x2.vemotherboard.sp810_sysctrl`
Only EB relevant functionalities are fully implemented.
Type: `SP810_SysCtrl`.

`FVP_VE_Cortex_R8x2.vemotherboard.sp810_sysctrl.clkdiv_clk0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x2.vemotherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x2.vemotherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x2.vemotherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x2.vemotherboard.terminal_4

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R8x2.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_VE_Cortex_R8x2.vemotherboard.ve_sysregs

Type: `VE_SysRegs`.

FVP_VE_Cortex_R8x2.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_VE_Cortex_R8x2.vemotherboard.vis

Display window for VE using Visualisation library.
Type: `VEVisualisation`.

FVP_VE_Cortex_R8x2.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R8x2.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x2.vemotherboard.vis.recorder.recordingDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

9.28 FVP_VE_Cortex-R8x4

FVP_VE_Cortex-R8x4 contains the following instances:

FVP_VE_Cortex-R8x4 instances

FVP_VE_Cortex_R8x4

Top level component of the Cortex_R8x4 Versatile Express inspired model.
Type: FVP_VE_Cortex_R8x4.

FVP_VE_Cortex_R8x4.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.daughterboard

Cortex_R8x4 DaughterBoard for Versatile Express.
Type: VEDaughterBoardCortex_R8x4.

FVP_VE_Cortex_R8x4.daughterboard.core

ARM CORTEXR8 Cluster CT model.
Type: Cluster_ARM_Cortex-R8.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu0

ARM CORTEXR8 CT model.
Type: ARM_Cortex-R8.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu0.l1dcache

PV Cache.

Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu0.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu1

ARM CORTEXR8 CT model.
Type: ARM_Cortex-R8.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu1.l1dcache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu1.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu2

ARM CORTEXR8 CT model.
Type: ARM_Cortex-R8.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu2.l1dcache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu2.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu3

ARM CORTEXR8 CT model.
Type: ARM_Cortex-R8.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu3.l1dcache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.core.cpu3.l1icache

PV Cache.
Type: PVCache.

FVP_VE_Cortex_R8x4.daughterboard.dram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R8x4.daughterboard.exclusive_monitor

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_VE_Cortex_R8x4.daughterboard.periph_clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.daughterboard.pl310_l2cc`
ARM PrimeCell Level 2 Cache Controller (PL310).
Type: `PL310_L2CC`.

`FVP_VE_Cortex_R8x4.daughterboard.veinterruptmapper`
Interrupt Mapping peripheral (non-cascaded).
Type: `VEInterruptMapper`.

`FVP_VE_Cortex_R8x4.vemotherboard`
Model inspired by the ARM Versatile Express Motherboard for R profile.
Type: `VEMotherBoardR`.

`FVP_VE_Cortex_R8x4.vemotherboard.audioout`
SDL based Audio Output for PL041_AACI.
Type: `AudioOut_SDL`.

`FVP_VE_Cortex_R8x4.vemotherboard.clock100Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.vemotherboard.clock24MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.vemotherboard.clock35MHz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.vemotherboard.clock50Hz`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.vemotherboard.clockCLCD`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.vemotherboard.dummy_CF`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_VE_Cortex_R8x4.vemotherboard.dummy_ram`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R8x4.vemotherboard.dummy_usb`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_VE_Cortex_R8x4.vemotherboard.flash0`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_R8x4.vemotherboard.flash1`
Intel Strata Flash J3 LISA+ model.
Type: `IntelStrataFlashJ3`.

`FVP_VE_Cortex_R8x4.vemotherboard.flashloader0`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_R8x4.vemotherboard.flashloader1`
A device that can preload a gzipped image into flash at startup.
Type: `FlashLoader`.

`FVP_VE_Cortex_R8x4.vemotherboard.hostbridge`
Host Socket Interface Component.
Type: `HostBridge`.

`FVP_VE_Cortex_R8x4.vemotherboard.mmc`
Generic Multimedia Card.
Type: `MMC`.

`FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart0`
ARM PrimeCell UART(PL011).
Type: `PL011_Uart`.

`FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart0.clk_divider`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart1`
ARM PrimeCell UART(PL011).

Type: PL011_Uart.

FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart2

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart3

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart3.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart4

ARM PrimeCell UART(PL011).
Type: PL011_Uart.

FVP_VE_Cortex_R8x4.vemotherboard.pl011_uart4.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.pl031_rtc

ARM PrimeCell Real Time Clock(PL031).
Type: PL031_RTC.

FVP_VE_Cortex_R8x4.vemotherboard.pl041_aaci

ARM PrimeCell Advanced Audio CODEC Interface(PL041).
Type: PL041_AACI.

FVP_VE_Cortex_R8x4.vemotherboard.pl050_kmi0

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).

Type: PL050_KMI.

FVP_VE_Cortex_R8x4.vemotherboard.pl050_kmi0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.pl050_kmi1

ARM PrimeCell PS2 Keyboard/Mouse Interface(PL050).
Type: PL050_KMI.

FVP_VE_Cortex_R8x4.vemotherboard.pl050_kmi1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.pl111_clcd

ARM PrimeCell Color LCD Controller(PL111).
Type: PL111_CLCD.

FVP_VE_Cortex_R8x4.vemotherboard.pl111_clcd.pl11x_clcd

Internal component used by PL110 and PL111 CLCD controllers.
Type: PL11x_CLCD.

FVP_VE_Cortex_R8x4.vemotherboard.pl111_clcd.pl11x_clcd.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_VE_Cortex_R8x4.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_VE_Cortex_R8x4.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_VE_Cortex_R8x4.vemotherboard.pl111_clcd.pl11x_clcd.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_VE_Cortex_R8x4.vemotherboard.pl180_mci

ARM PrimeCell Multimedia Card Interface (PL180).
Type: PL180_MCI.

FVP_VE_Cortex_R8x4.vemotherboard.ps2keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Keyboard.

FVP_VE_Cortex_R8x4.vemotherboard.ps2mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050_KMI component.
Type: PS2Mouse.

FVP_VE_Cortex_R8x4.vemotherboard.psram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R8x4.vemotherboard.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_VE_Cortex_R8x4.vemotherboard.sp805_wdog

ARM Watchdog Module(SP805).
Type: SP805_Watchdog.

FVP_VE_Cortex_R8x4.vemotherboard.sp810_sysctrl

Only EB relevant functionalities are fully implemented.
Type: SP810_SysCtrl.

FVP_VE_Cortex_R8x4.vemotherboard.sp810_sysctrl.clkdiv_clk0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.sp810_sysctrl.clkdiv_clk1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.sp810_sysctrl.clkdiv_clk2

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x4.vemotherboard.sp810_sysctrl.clkdiv_clk3

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x4.vemotherboard.terminal_0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x4.vemotherboard.terminal_1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x4.vemotherboard.terminal_2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x4.vemotherboard.terminal_3

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x4.vemotherboard.terminal_4

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_VE_Cortex_R8x4.vemotherboard.timer_0_1

ARM Dual-Timer Module(SP804).
Type: `SP804_Timer`.

FVP_VE_Cortex_R8x4.vemotherboard.timer_0_1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x4.vemotherboard.timer_0_1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_VE_Cortex_R8x4.vemotherboard.timer_0_1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x4.vemotherboard.timer_0_1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x4.vemotherboard.timer_2_3

ARM Dual-Timer Module(SP804).
Type: SP804_Timer.

FVP_VE_Cortex_R8x4.vemotherboard.timer_2_3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.timer_2_3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_VE_Cortex_R8x4.vemotherboard.timer_2_3.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x4.vemotherboard.timer_2_3.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_VE_Cortex_R8x4.vemotherboard.ve_sysregs

Type: VE_SysRegs.

FVP_VE_Cortex_R8x4.vemotherboard.video_ram

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_VE_Cortex_R8x4.vemotherboard.vis

Display window for VE using Visualisation library.
Type: VEVisualisation.

FVP_VE_Cortex_R8x4.vemotherboard.vis.recorder

Event recorder component for visualisation component (allows to playback and record interactive GUI sessions).
Type: VisEventRecorder.

FVP_VE_Cortex_R8x4.vemotherboard.vis.recorder.playbackDivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_VE_Cortex_R8x4.vemotherboard.vis.recorder.recordingDivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

10 MPS2 Platform FVPs

This chapter lists the MPS2 Platform FVPs and the instances in them.

For the MPS2 memory maps, see MPS2 - memory maps in the Fast Models Reference Manual.

10.1 FVP_MPS2_AEMv8M

FVP_MPS2_AEMv8M contains the following instances:

FVP_MPS2_AEMv8M instances

FVP_MPS2_AEMv8M

Type: `FVP_MPS2_AEMv8M`.

FVP_MPS2_AEMv8M.aborting_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_AEMv8M.clk25Mhz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.cpu0

ARM AEMv8M CT model.
Type: `ARM_AEMv8M`.

FVP_MPS2_AEMv8M.cpu1

ARM AEMv8M CT model.
Type: `ARM_AEMv8M`.

FVP_MPS2_AEMv8M.fvp_mps2

MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_AEMv8M.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_AEMv8M.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_AEMv8M.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_AEMv8M.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_AEMv8M.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_AEMv8M.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_AEMv8M.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_AEMv8M.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_AEMv8M.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_AEMv8M.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.UART2

ARM CMSDK UART Module.

Type: `CMSDK_UART`.

`FVP_MPS2_AEMv8M.fvp_mps2.UART2.clk_divider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_AEMv8M.fvp_mps2.VGA_interface`

VGA display interface between main bus and visualisation.
Type: `MPS2_VGA`.

`FVP_MPS2_AEMv8M.fvp_mps2.ahb_ppc_iotss_expansion0`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_AEMv8M.fvp_mps2.ahb_ppc_iotss_expansion1`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_AEMv8M.fvp_mps2.apb_ppc_iotss_expansion0`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_AEMv8M.fvp_mps2.apb_ppc_iotss_expansion1`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_AEMv8M.fvp_mps2.apb_ppc_iotss_expansion2`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_AEMv8M.fvp_mps2.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_AEMv8M.fvp_mps2.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_AEMv8M.fvp_mps2.cmsdk_sysctrl`

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

`FVP_MPS2_AEMv8M.fvp_mps2.cmsdk_watchdog`

ARM Watchdog Module.

Type: `CMSDK_Watchdog`.

`FVP_MPS2_AEMv8M.fvp_mps2.cpu_wait_or_gate_0`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_AEMv8M.fvp_mps2.cpu_wait_or_gate_1`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_AEMv8M.fvp_mps2.dbgen_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_AEMv8M.fvp_mps2.dma0`

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_AEMv8M.fvp_mps2.dma0.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_MPS2_AEMv8M.fvp_mps2.dma0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_MPS2_AEMv8M.fvp_mps2.dma0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_MPS2_AEMv8M.fvp_mps2.dma0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_AEMv8M.fvp_mps2.dma0_idau_labeller`

Type: `LabellerIdauSecurity`.

`FVP_MPS2_AEMv8M.fvp_mps2.dma0_securitymodifier`

Type: `SecurityModifier`.

FVP_MPS2_AEMv8M.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_AEMv8M.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_AEMv8M.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_AEMv8M.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_AEMv8M.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_AEMv8M.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_AEMv8M.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_AEMv8M.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_AEMv8M.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_AEMv8M.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_AEMv8M.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_AEMv8M.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.dma2_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_AEMv8M.fvp_mps2.dma2_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_AEMv8M.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_AEMv8M.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_AEMv8M.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_AEMv8M.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_AEMv8M.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_AEMv8M.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_AEMv8M.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_AEMv8M.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_AEMv8M.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_AEMv8M.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_AEMv8M.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_AEMv8M.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_AEMv8M.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_AEMv8M.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.gpio_0_or_2

Or Gate.
Type: `OrGate`.

FVP_MPS2_AEMv8M.fvp_mps2.gpio_1_or_3

Or Gate.
Type: `OrGate`.

FVP_MPS2_AEMv8M.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_MPS2_AEMv8M.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_AEMv8M.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_AEMv8M.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_audio

MPS2 Audio.
Type: `MPS2_Audio`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.

Type: `MPS2_SecureCtrl`.

`FVP_MPS2_AEMv8M.fvp_mps2.mps2_timer0`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_AEMv8M.fvp_mps2.mps2_timer0.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_AEMv8M.fvp_mps2.mps2_timer0.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_AEMv8M.fvp_mps2.mps2_timer1`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_AEMv8M.fvp_mps2.mps2_timer1.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_AEMv8M.fvp_mps2.mps2_timer1.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_AEMv8M.fvp_mps2.mps2_visualisation`

Display window for MPS2 using Visualisation library.
Type: `MPS2_Visualisation`.

`FVP_MPS2_AEMv8M.fvp_mps2.niden_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_AEMv8M.fvp_mps2.nmi_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_AEMv8M.fvp_mps2.pl022_ssp_mps2`

ARM PrimeCell Synchronous Serial Port(PL022).
Type: `PL022_SSP_MPS2`.

`FVP_MPS2_AEMv8M.fvp_mps2.pl022_ssp_mps2.prescaler`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_AEMv8M.fvp_mps2.platform_bus_switch
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_AEMv8M.fvp_mps2.platform_switch_dma0
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_AEMv8M.fvp_mps2.platform_switch_dma1
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_AEMv8M.fvp_mps2.signal_router
Signal router.
Type: SignalRouter.

FVP_MPS2_AEMv8M.fvp_mps2.smsc_91c111
SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_AEMv8M.fvp_mps2.spiden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_AEMv8M.fvp_mps2.spniden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_AEMv8M.fvp_mps2.sse200
SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cmsdk_dualtimer.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cordio_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cpu0core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.iotss_cpuidentity

IoT Subsystem CPU_IDENTITY registers.
Type: `IoTSS_CPUIdentity`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.iotss_internal_sram0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.iotss_internal_sram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: `IoTSS_SystemControl`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: `IoTSS_SystemInfo`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.

Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer

Type: `SVOS_DualTimer`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_AEMv8M.fvp_mps2.switch_PSRAM_M7

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_AEMv8M.fvp_mps2.switch_svos_dualtimer
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_AEMv8M.fvp_mps2.telnetterminal0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_AEMv8M.fvp_mps2.telnetterminal1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_AEMv8M.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_AEMv8M.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: MPS2_TouchScreen.

FVP_MPS2_AEMv8M.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: OrGate.

10.2 FVP_MPS2_Cortex-M0

FVP_MPS2_Cortex-M0 contains the following instances:

FVP_MPS2_Cortex-M0 instances

FVP_MPS2_Cortex_M0
Type: FVP_MPS2_Cortex_M0.

FVP_MPS2_Cortex_M0.armcortexm0ct
ARM CORTEXM0 CT model.
Type: ARM_Cortex-M0.

FVP_MPS2_Cortex_M0.clk25Mhz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.clk25khz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2
MPS2 DUT.
Type: FVP_MPS2.

FVP_MPS2_Cortex_M0.fvp_mps2.GPIO0
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M0.fvp_mps2.GPIO1
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M0.fvp_mps2.GPIO2
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M0.fvp_mps2.GPIO3
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M0.fvp_mps2.GPIO_connection_test
Type: GPIO_Connection_Test.

FVP_MPS2_Cortex_M0.fvp_mps2.GPIO_connection_test.GPIO0_port_trans
Type: GPIO_Port_Transfer.

FVP_MPS2_Cortex_M0.fvp_mps2.GPIO_connection_test.GPIO1_port_test
Type: GPIO1_Connection_Test.

FVP_MPS2_Cortex_M0.fvp_mps2.PSRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.PSRAM_M7
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.UART0
ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M0.fvp_mps2.UART0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2.UART1
ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M0.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M0.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: `MPS2_VGA`.

FVP_MPS2_Cortex_M0.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.cmsdk_sysctrl`
Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

`FVP_MPS2_Cortex_M0.fvp_mps2.cmsdk_watchdog`
ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

`FVP_MPS2_Cortex_M0.fvp_mps2.cpu_wait_or_gate_0`
Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.cpu_wait_or_gate_1`
Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dbgen_or_gate`
Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma0`
ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma0.timer`
A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma0.timer.timer`
A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma0.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M0.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M0.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M0.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M0.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M0.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M0.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M0.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M0.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M0.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma2_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma2_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M0.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a

proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma3.timer.timer.thread`
A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma3.timer.timer.thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma3_idau_labeller`
Type: `LabellerIdauSecurity`.

`FVP_MPS2_Cortex_M0.fvp_mps2.dma3_securitymodifier`
Type: `SecurityModifier`.

`FVP_MPS2_Cortex_M0.fvp_mps2.exclusive_monitor_psram`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_Cortex_M0.fvp_mps2.exclusive_monitor_psram_iotss`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_Cortex_M0.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M0.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M0.fvp_mps2.exclusive_monitor_zbtsram1`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_Cortex_M0.fvp_mps2.exclusive_monitor_zbtsram2`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_Cortex_M0.fvp_mps2.exclusive_squasher`
Squashes the exclusive attribute on bus transactions.
Type: `PVBusExclusiveSquasher`.

`FVP_MPS2_Cortex_M0.fvp_mps2.expansion_warning_memory`
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_Cortex_M0.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: `FPGA_SysCtrl`.

FVP_MPS2_Cortex_M0.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M0.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.gpio_0_or_2

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.gpio_1_or_3

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_MPS2_Cortex_M0.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M0.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M0.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_audio

MPS2 Audio.
Type: `MPS2_Audio`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_Cortex_M0.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.

Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_secure_control_register_block`

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_timer0`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_timer0.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_timer0.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_timer1`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_timer1.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_timer1.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.mps2_visualisation`

Display window for MPS2 using Visualisation library.
Type: `MPS2_Visualisation`.

`FVP_MPS2_Cortex_M0.fvp_mps2.niden_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.nmi_or_gate`

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: `PL022_SSP_MPS2`.

FVP_MPS2_Cortex_M0.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M0.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M0.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M0.fvp_mps2.signal_router

Signal router.
Type: `SignalRouter`.

FVP_MPS2_Cortex_M0.fvp_mps2.smsc_91c111

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_MPS2_Cortex_M0.fvp_mps2.spiden_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.spniden_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200

SSE-200 subsystem.
Type: `SSE200`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.acg_cpu0

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.acg_cpu1

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.acg_sram0

IoT Subsystem Access Control Gate.

Type: `IoTSS_AccessControlGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.acg_sram1`

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.acg_sram2`

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.acg_sram3`

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.apb_ppc_iotss_subsystem0`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.apb_ppc_iotss_subsystem1`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.bus_error_warning_memory`

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.clock32kHz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cmsdk_dualtimer`

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cpu0dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cpu1core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.cpu1dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.crypto_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2

Global exclusive monitor.

Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.iotss_cpuidentity

IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.iotss_internal_sram0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.iotss_internal_sram1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.ram0_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.ram1_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.ram2_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.ram3_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.s32k_timer
ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.s32k_timer.clk_div
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.s32k_timer.counter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.s32k_watchdog
ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.secure_control_register_block
MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.secure_watchdog
ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.signal_router
Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.sys_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.timer0
ARM Timer Module.

Type: CMSDK_Timer.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer2`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer3`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0.fvp_mps2.switch_PSRAM_M7`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M0.fvp_mps2.switch_svos_dualtimer`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M0.fvp_mps2.telnetterminal0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M0.fvp_mps2.telnetterminal1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M0.fvp_mps2.telnetterminal2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M0.fvp_mps2.touchscreen_interface`
MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

`FVP_MPS2_Cortex_M0.fvp_mps2.uart_overflows_or_gate`
Or Gate.
Type: `OrGate`.

10.3 FVP_MPS2_Cortex-M0plus

FVP_MPS2_Cortex-M0plus contains the following instances:

FVP_MPS2_Cortex-M0plus instances

`FVP_MPS2_Cortex_M0plus`
Type: `FVP_MPS2_Cortex_M0plus`.

`FVP_MPS2_Cortex_M0plus.armcortexm0plusct`
ARM CORTEXM0+ CT model.

Type: `ARM_Cortex-M0+`.

`FVP_MPS2_Cortex_M0plus.clk25Mhz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.clk25khz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2`

MPS2 DUT.
Type: `FVP_MPS2`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.GPIO0`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.GPIO1`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.GPIO2`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.GPIO3`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.GPIO_connection_test`

Type: `GPIO_Connection_Test`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.GPIO_connection_test.GPIO0_port_trans`

Type: `GPIO_Port_Transfer`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.GPIO_connection_test.GPIO1_port_test`

Type: `GPIO1_Connection_Test`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.PSRAM`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.PSRAM_M7`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: `MPS2_VGA`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.cmsdk_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.cpu_wait_or_gate_0

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.cpu_wait_or_gate_1

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dbgen_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma0_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma0_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma3_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.dma3_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.

Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M0plus.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M0plus.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M0plus.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.gpio_0_or_2

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M0plus.fvp_mps2.gpio_1_or_3

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M0plus.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: MPS2_LCD.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_Cortex_M0plus.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M0plus.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M0plus.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_Cortex_M0plus.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_Cortex_M0plus.fvp_mps2.spiden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M0plus.fvp_mps2.spniden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200
SSE-200 subsystem.
Type: `SSE200`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.acg_sram3
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.apb_ppc_iotss_subsystem0
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.apb_ppc_iotss_subsystem1
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.bus_error_warning_memory
Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.clock32kHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cpu0dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cpu1core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.cpu1dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.crypto_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.iotss_internal_sram2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.iotss_internal_sram3
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.iotss_systemcontrol
IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.iotss_systeminfo
IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mem_switch_internal_sram
Allow transactions to be routed arbitrarily.

Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.signal_router

Signal router.

Type: `SignalRouter`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.sys_ppu`

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.timer0`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.timer0.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.timer0.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.timer1`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.timer1.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.sse200.timer1.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.ssram1`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.ssram2`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.stub0`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.stub1`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M0plus.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer2`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer3`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.switch_PSRAM_M7`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.switch_svos_dualtimer`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.telnetterminal0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.telnetterminal1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.telnetterminal2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.touchscreen_interface`
MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

`FVP_MPS2_Cortex_M0plus.fvp_mps2.uart_overflows_or_gate`
Or Gate.
Type: `OrGate`.

10.4 FVP_MPS2_Cortex-M23

FVP_MPS2_Cortex-M23 contains the following instances:

FVP_MPS2_Cortex-M23 instances

FVP_MPS2_Cortex_M23

Type: FVP_MPS2_Cortex_M23.

FVP_MPS2_Cortex_M23.clk25Mhz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.cpu0

ARM CORTEXM23 CT model.
Type: ARM_Cortex-M23.

FVP_MPS2_Cortex_M23.cpu1

ARM CORTEXM23 CT model.
Type: ARM_Cortex-M23.

FVP_MPS2_Cortex_M23.fvp_mps2

MPS2 DUT.
Type: FVP_MPS2.

FVP_MPS2_Cortex_M23.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M23.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M23.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M23.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M23.fvp_mps2.GPIO_connection_test

Type: GPIO_Connection_Test.

FVP_MPS2_Cortex_M23.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: GPIO_Port_Transfer.

FVP_MPS2_Cortex_M23.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: GPIO1_Connection_Test.

FVP_MPS2_Cortex_M23.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M23.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M23.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M23.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: MPS2_VGA.

FVP_MPS2_Cortex_M23.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M23.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M23.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

FVP_MPS2_Cortex_M23.fvp_mps2.cmsdk_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M23.fvp_mps2.cpu_wait_or_gate_0

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.cpu_wait_or_gate_1

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.dbgen_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M23.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M23.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M23.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M23.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M23.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M23.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M23.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M23.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma1_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma1_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M23.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M23.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M23.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M23.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M23.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M23.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M23.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M23.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M23.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.exclusive_monitor_psram_iotss
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.exclusive_monitor_zbtsram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.exclusive_monitor_zbtsram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M23.fvp_mps2.expansion_warning_memory
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M23.fvp_mps2.fpga_sysctrl
FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M23.fvp_mps2.fpga_sysctrl.callBack100HzCounter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.fpga_sysctrl.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.gpio_0_or_2
Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M23.fvp_mps2.gpio_1_or_3
Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M23.fvp_mps2.hostbridge
Host Socket Interface Component.

Type: HostBridge.

FVP_MPS2_Cortex_M23.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M23.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M23.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M23.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_Cortex_M23.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M23.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M23.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_Cortex_M23.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.signal_router
Signal router.
Type: `SignalRouter`.

FVP_MPS2_Cortex_M23.fvp_mps2.smsc_91c111
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_MPS2_Cortex_M23.fvp_mps2.spiden_or_gate
Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.spniden_or_gate
Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200
SSE-200 subsystem.
Type: `SSE200`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.acg_sram3
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.apb_ppc_iotss_subsystem0
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.apb_ppc_iotss_subsystem1
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.

Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.sse200.timer1.counter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M23.fvp_mps2.ssram1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.ssram2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.stub0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.stub1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.stub_i2c1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.stub_i2s
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.stub_spi0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.stub_spi2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer
Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer0
ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer1`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer2`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer3`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M23.fvp_mps2.switch_PSRAM_M7`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M23.fvp_mps2.switch_svos_dualtimer`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M23.fvp_mps2.telnetterminal0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M23.fvp_mps2.telnetterminal1`
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M23.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_Cortex_M23.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: MPS2_TouchScreen.

FVP_MPS2_Cortex_M23.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: OrGate.

10.5 FVP_MPS2_Cortex-M3

FVP_MPS2_Cortex-M3 contains the following instances:

FVP_MPS2_Cortex-M3 instances

FVP_MPS2_Cortex_M3
Type: FVP_MPS2_Cortex_M3.

FVP_MPS2_Cortex_M3.armcortexm3ct
ARM CORTEXM3 CT model.
Type: ARM_Cortex-M3.

FVP_MPS2_Cortex_M3.clk25Mhz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M3.clk25khz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M3.fvp_mps2
MPS2 DUT.
Type: FVP_MPS2.

FVP_MPS2_Cortex_M3.fvp_mps2.GPIO0
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M3.fvp_mps2.GPIO1
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M3.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M3.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M3.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_Cortex_M3.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_Cortex_M3.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_Cortex_M3.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M3.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M3.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M3.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: `MPS2_VGA`.

FVP_MPS2_Cortex_M3.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

FVP_MPS2_Cortex_M3.fvp_mps2.cmsdk_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M3.fvp_mps2.cpu_wait_or_gate_0

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M3.fvp_mps2.cpu_wait_or_gate_1

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M3.fvp_mps2.dbgen_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M3.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M3.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M3.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M3.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M3.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M3.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M3.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M3.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).

Type: PL080_DMAC.

FVP_MPS2_Cortex_M3.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M3.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M3.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M3.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M3.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M3.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M3.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M3.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M3.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma2_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma2_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M3.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M3.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M3.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M3.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M3.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M3.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M3.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M3.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.gpio_0_or_2

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M3.fvp_mps2.gpio_1_or_3

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M3.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_MPS2_Cortex_M3.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M3.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M3.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M3.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M3.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_audio

MPS2 Audio.
Type: `MPS2_Audio`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M3.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.

Type: `MPS2_SecureCtrl`.

`FVP_MPS2_Cortex_M3.fvp_mps2.mps2_timer0`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M3.fvp_mps2.mps2_timer0.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M3.fvp_mps2.mps2_timer0.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M3.fvp_mps2.mps2_timer1`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M3.fvp_mps2.mps2_timer1.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M3.fvp_mps2.mps2_timer1.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M3.fvp_mps2.mps2_visualisation`

Display window for MPS2 using Visualisation library.
Type: `MPS2_Visualisation`.

`FVP_MPS2_Cortex_M3.fvp_mps2.niden_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M3.fvp_mps2.nmi_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M3.fvp_mps2.pl022_ssp_mps2`

ARM PrimeCell Synchronous Serial Port(PL022).
Type: `PL022_SSP_MPS2`.

`FVP_MPS2_Cortex_M3.fvp_mps2.pl022_ssp_mps2.prescaler`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_Cortex_M3.fvp_mps2.platform_bus_switch
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.platform_switch_dma0
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.platform_switch_dma1
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.signal_router
Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.smsc_91c111
SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_Cortex_M3.fvp_mps2.spiden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M3.fvp_mps2.spniden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200
SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cmsdk_dualtimer.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cordio_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cpu0core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.iotss_cpuidentity

IoT Subsystem CPU_IDENTITY registers.
Type: `IoTSS_CPUIdentity`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.iotss_internal_sram0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.iotss_internal_sram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: `IoTSS_SystemControl`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: `IoTSS_SystemInfo`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.

Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram1`
IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram2`
IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram3`
IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.nmi_or_gate`
Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.nonsecure_watchdog`
ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.ram0_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.ram1_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M3.fvp_mps2.sse200.ram2_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer

Type: `SVOS_DualTimer`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M3.fvp_mps2.switch_PSRAM_M7

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M3.fvp_mps2.switch_svos_dualtimer
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M3.fvp_mps2.telnetterminal0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_Cortex_M3.fvp_mps2.telnetterminal1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_Cortex_M3.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_Cortex_M3.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: MPS2_TouchScreen.

FVP_MPS2_Cortex_M3.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: OrGate.

10.6 FVP_MPS2_Cortex-M33

FVP_MPS2_Cortex-M33 contains the following instances:

FVP_MPS2_Cortex-M33 instances

FVP_MPS2_Cortex_M33
Type: FVP_MPS2_Cortex_M33.

FVP_MPS2_Cortex_M33.clk25Mhz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.clk25khz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.cpu0
ARM CORTEXM33 CT model.
Type: ARM_Cortex-M33.

FVP_MPS2_Cortex_M33.cpu1
ARM CORTEXM33 CT model.
Type: ARM_Cortex-M33.

FVP_MPS2_Cortex_M33.fvp_mps2
MPS2 DUT.
Type: FVP_MPS2.

FVP_MPS2_Cortex_M33.fvp_mps2.GPIO0
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M33.fvp_mps2.GPIO1
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M33.fvp_mps2.GPIO2
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M33.fvp_mps2.GPIO3
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M33.fvp_mps2.GPIO_connection_test
Type: GPIO_Connection_Test.

FVP_MPS2_Cortex_M33.fvp_mps2.GPIO_connection_test.GPIO0_port_trans
Type: GPIO_Port_Transfer.

FVP_MPS2_Cortex_M33.fvp_mps2.GPIO_connection_test.GPIO1_port_test
Type: GPIO1_Connection_Test.

FVP_MPS2_Cortex_M33.fvp_mps2.PSRAM
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.PSRAM_M7
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.UART0
ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M33.fvp_mps2.UART0.clk_divider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M33.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M33.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: `MPS2_VGA`.

FVP_MPS2_Cortex_M33.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

FVP_MPS2_Cortex_M33.fvp_mps2.cmsdk_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M33.fvp_mps2.cpu_wait_or_gate_0

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.cpu_wait_or_gate_1

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.dbgen_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: SchedulerThread.

FVP_MPS2_Cortex_M33.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M33.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M33.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M33.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M33.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M33.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M33.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M33.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M33.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).

Type: PL080_DMAC.

FVP_MPS2_Cortex_M33.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M33.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M33.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M33.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M33.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M33.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M33.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M33.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma3_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M33.fvp_mps2.dma3_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M33.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M33.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M33.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M33.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M33.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M33.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M33.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: `PVBusExclusiveSquasher`.

FVP_MPS2_Cortex_M33.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M33.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M33.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.gpio_0_or_2

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M33.fvp_mps2.gpio_1_or_3

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M33.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_Cortex_M33.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M33.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M33.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M33.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: `MPS2_Visualisation`.

FVP_MPS2_Cortex_M33.fvp_mps2.niden_or_gate

Or Gate.

Type: OrGate.

FVP_MPS2_Cortex_M33.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M33.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_Cortex_M33.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_Cortex_M33.fvp_mps2.spiden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M33.fvp_mps2.spniden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200

SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.acg_cpu0

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.acg_cpu1

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.acg_sram0

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.acg_sram1

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.acg_sram2

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cordio_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cpu0core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.

Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.iotss_cpuidentity

IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.nonsecure_watchdog
ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.ram0_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.ram1_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.ram2_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.ram3_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.s32k_timer
ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.s32k_timer.clk_div
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.s32k_timer.counter
Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.s32k_watchdog
ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.secure_control_register_block
MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.secure_watchdog
ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.signal_router
Signal router.
Type: `SignalRouter`.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.sys_ppu
ARM Power Policy Unit (PPU) architectural model.

Type: PPUv0.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M33.fvp_mps2.switch_PSRAM_M7

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M33.fvp_mps2.switch_svos_dualtimer

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M33.fvp_mps2.telnetterminal0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M33.fvp_mps2.telnetterminal1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M33.fvp_mps2.telnetterminal2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M33.fvp_mps2.touchscreen_interface

MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

FVP_MPS2_Cortex_M33.fvp_mps2.uart_overflows_or_gate

Or Gate.
Type: `OrGate`.

10.7 FVP_MPS2_Cortex-M35P

FVP_MPS2_Cortex-M35P contains the following instances:

FVP_MPS2_Cortex-M35P instances

FVP_MPS2_Cortex_M35P

Type: `FVP_MPS2_Cortex_M35P`.

FVP_MPS2_Cortex_M35P.clk25Mhz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.cpu0

ARM CORTEXM35P CT model.
Type: `ARM_Cortex-M35P`.

FVP_MPS2_Cortex_M35P.cpu1

ARM CORTEXM35P CT model.
Type: `ARM_Cortex-M35P`.

FVP_MPS2_Cortex_M35P.fvp_mps2

MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_Cortex_M35P.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M35P.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M35P.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M35P.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M35P.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_Cortex_M35P.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_Cortex_M35P.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M35P.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M35P.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M35P.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: MPS2_VGA.

FVP_MPS2_Cortex_M35P.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M35P.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M35P.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.

Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.apb_ppc_iotss_expansion1`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.apb_ppc_iotss_expansion2`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.cmsdk_sysctrl`

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.cmsdk_watchdog`

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.cpu_wait_or_gate_0`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.cpu_wait_or_gate_1`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dbgen_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma0`

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma0.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for

the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma0_idau_labeller`

Type: `LabellerIdauSecurity`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma0_securitymodifier`

Type: `SecurityModifier`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma1`

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma1.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.dma1.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M35P.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M35P.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.

Type: PVBusRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M35P.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M35P.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M35P.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M35P.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.gpio_0_or_2

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.gpio_1_or_3

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_Cortex_M35P.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_audio

MPS2 Audio.
Type: `MPS2_Audio`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: MPS2_LCD.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M35P.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_Cortex_M35P.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_Cortex_M35P.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M35P.fvp_mps2.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_Cortex_M35P.fvp_mps2.spiden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.spniden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200

SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.acg_cpu0

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.acg_cpu1

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.acg_sram0

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.acg_sram1

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.acg_sram2

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cpu0dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cpu1core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.cpu1dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.iotss_internal_sram2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.iotss_internal_sram3
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.iotss_systemcontrol
IoT Subsystem System Control registers.

Type: `IoTSS_SystemControl`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.iotss_systeminfo`

IoT Subsystem System Information registers.
Type: `IoTSS_SystemInfo`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mem_switch_internal_sram`

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mem_switch_internal_sram_mpc`

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mem_switch_ppu`

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mhu0`

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mhu1`

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram0`

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram1`

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram2`

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.signal_router

Signal router.
Type: `SignalRouter`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M35P.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M35P.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M35P.fvp_mps2.switch_PSRAM_M7

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M35P.fvp_mps2.switch_svos_dualtimer

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M35P.fvp_mps2.telnetterminal0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M35P.fvp_mps2.telnetterminal1

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M35P.fvp_mps2.telnetterminal2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M35P.fvp_mps2.touchscreen_interface

MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

FVP_MPS2_Cortex_M35P.fvp_mps2.uart_overflows_or_gate

Or Gate.
Type: `OrGate`.

10.8 FVP_MPS2_Cortex-M4

FVP_MPS2_Cortex-M4 contains the following instances:

FVP_MPS2_Cortex-M4 instances

`FVP_MPS2_Cortex_M4`

Type: `FVP_MPS2_Cortex_M4`.

`FVP_MPS2_Cortex_M4.armcortexm4ct`

ARM CORTEXM4 CT model.
Type: `ARM_Cortex-M4`.

`FVP_MPS2_Cortex_M4.clk25Mhz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.clk25khz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2`

MPS2 DUT.
Type: `FVP_MPS2`.

`FVP_MPS2_Cortex_M4.fvp_mps2.GPIO0`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M4.fvp_mps2.GPIO1`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M4.fvp_mps2.GPIO2`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M4.fvp_mps2.GPIO3`

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

`FVP_MPS2_Cortex_M4.fvp_mps2.GPIO_connection_test`

Type: `GPIO_Connection_Test`.

FVP_MPS2_Cortex_M4.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_Cortex_M4.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_Cortex_M4.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.

Type: `RAMDevice`.

FVP_MPS2_Cortex_M4.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.

Type: `RAMDevice`.

FVP_MPS2_Cortex_M4.fvp_mps2.UART0

ARM CMSDK UART Module.

Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M4.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.UART1

ARM CMSDK UART Module.

Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M4.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.UART2

ARM CMSDK UART Module.

Type: `CMSDK_UART`.

FVP_MPS2_Cortex_M4.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.

Type: `MPS2_VGA`.

FVP_MPS2_Cortex_M4.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.

Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M4.fvp_mps2.ahb_ppc_iotss_expansion1`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M4.fvp_mps2.apb_ppc_iotss_expansion0`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M4.fvp_mps2.apb_ppc_iotss_expansion1`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M4.fvp_mps2.apb_ppc_iotss_expansion2`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M4.fvp_mps2.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.cmsdk_sysctrl`

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

`FVP_MPS2_Cortex_M4.fvp_mps2.cmsdk_watchdog`

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

`FVP_MPS2_Cortex_M4.fvp_mps2.cpu_wait_or_gate_0`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M4.fvp_mps2.cpu_wait_or_gate_1`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M4.fvp_mps2.dbgen_or_gate`

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M4.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M4.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M4.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M4.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M4.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M4.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M4.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M4.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma1_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma1_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M4.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M4.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M4.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M4.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M4.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M4.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M4.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M4.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M4.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.exclusive_monitor_psram_iotss
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.exclusive_monitor_zbtsram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.exclusive_monitor_zbtsram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M4.fvp_mps2.expansion_warning_memory
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M4.fvp_mps2.fpga_sysctrl
FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M4.fvp_mps2.fpga_sysctrl.callBack100HzCounter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M4.fvp_mps2.fpga_sysctrl.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.gpio_0_or_2
Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M4.fvp_mps2.gpio_1_or_3
Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M4.fvp_mps2.hostbridge
Host Socket Interface Component.

Type: HostBridge.

FVP_MPS2_Cortex_M4.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M4.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M4.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M4.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M4.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_Cortex_M4.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M4.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M4.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_Cortex_M4.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.signal_router
Signal router.
Type: `SignalRouter`.

FVP_MPS2_Cortex_M4.fvp_mps2.smsc_91c111
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_MPS2_Cortex_M4.fvp_mps2.spiden_or_gate
Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.spniden_or_gate
Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200
SSE-200 subsystem.
Type: `SSE200`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.acg_sram3
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.apb_ppc_iotss_subsystem0
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.apb_ppc_iotss_subsystem1
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.

Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.sse200.timer1.counter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M4.fvp_mps2.ssram1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.ssram2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.stub0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.stub1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.stub_i2c1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.stub_i2s
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.stub_spi0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.stub_spi2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer
Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer0
ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer1`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer2`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer3`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M4.fvp_mps2.switch_PSRAM_M7`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M4.fvp_mps2.switch_svos_dualtimer`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M4.fvp_mps2.telnetterminal0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M4.fvp_mps2.telnetterminal1`
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M4.fvp_mps2.telnetterminal2

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_Cortex_M4.fvp_mps2.touchscreen_interface

MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

FVP_MPS2_Cortex_M4.fvp_mps2.uart_overflows_or_gate

Or Gate.
Type: `OrGate`.

10.9 FVP_MPS2_Cortex-M55

FVP_MPS2_Cortex-M55 contains the following instances:

FVP_MPS2_Cortex-M55 instances

FVP_MPS2_Cortex_M55

Type: `FVP_MPS2_Cortex_M55`.

FVP_MPS2_Cortex_M55.clk25Mhz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M55.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M55.core0_bus_gasket

Type: `STLBusGasket`.

FVP_MPS2_Cortex_M55.cpu0

ARM Cortex-M55 CT model.
Type: `ARM_Cortex-M55`.

FVP_MPS2_Cortex_M55.cpu1

ARM Cortex-M55 CT model.
Type: `ARM_Cortex-M55`.

FVP_MPS2_Cortex_M55.fvp_mps2

MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_Cortex_M55.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M55.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M55.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M55.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_Cortex_M55.fvp_mps2.GPIO_connection_test

Type: GPIO_Connection_Test.

FVP_MPS2_Cortex_M55.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: GPIO_Port_Transfer.

FVP_MPS2_Cortex_M55.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: GPIO1_Connection_Test.

FVP_MPS2_Cortex_M55.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M55.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M55.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M55.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M55.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M55.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: MPS2_VGA.

FVP_MPS2_Cortex_M55.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.

Type: `CMSDK_SysCtrl`.

`FVP_MPS2_Cortex_M55.fvp_mps2.cmsdk_watchdog`

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

`FVP_MPS2_Cortex_M55.fvp_mps2.cpu_wait_or_gate_0`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M55.fvp_mps2.cpu_wait_or_gate_1`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M55.fvp_mps2.dbgen_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M55.fvp_mps2.dma0`

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_Cortex_M55.fvp_mps2.dma0.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_MPS2_Cortex_M55.fvp_mps2.dma0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_MPS2_Cortex_M55.fvp_mps2.dma0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_MPS2_Cortex_M55.fvp_mps2.dma0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_Cortex_M55.fvp_mps2.dma0_idau_labeller`

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma0_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma1_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma1_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_Cortex_M55.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M55.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M55.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M55.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M55.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M55.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M55.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M55.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M55.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M55.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M55.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M55.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M55.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M55.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M55.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M55.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M55.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M55.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M55.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M55.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M55.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.gpio_0_or_2

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M55.fvp_mps2.gpio_1_or_3

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M55.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_Cortex_M55.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M55.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M55.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: MPS2_LCD.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_Cortex_M55.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M55.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M55.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).

Type: PL022_SSP_MPS2.

FVP_MPS2_Cortex_M55.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_Cortex_M55.fvp_mps2.spiden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M55.fvp_mps2.spniden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200

SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.acg_cpu0

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.acg_cpu1

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.acg_sram0

IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.acg_sram1

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.acg_sram2

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cordio_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cpu0core_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cpu0dbg_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cpu1core_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.cpu1dbg_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.crypto_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.dbg_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3`
Global exclusive monitor.

Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.iotss_cpuidentity

IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.ram1_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.ram2_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.ram3_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.s32k_timer
ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.s32k_timer.clk_div
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.s32k_timer.counter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.s32k_watchdog
ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.secure_control_register_block
MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.secure_watchdog
ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.signal_router
Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.sys_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.timer0
ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M55.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M55.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M55.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M55.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M55.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M55.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M55.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M55.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M55.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M55.fvp_mps2.switch_PSRAM_M7
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.switch_svos_dualtimer
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M55.fvp_mps2.telnetterminal0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_Cortex_M55.fvp_mps2.telnetterminal1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_Cortex_M55.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_Cortex_M55.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: MPS2_TouchScreen.

FVP_MPS2_Cortex_M55.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: OrGate.

10.10 FVP_MPS2_Cortex-M7

FVP_MPS2_Cortex-M7 contains the following instances:

FVP_MPS2_Cortex-M7 instances

FVP_MPS2_Cortex_M7
Type: FVP_MPS2_Cortex_M7.

FVP_MPS2_Cortex_M7.armcortexm7ct
ARM CORTEXM7 CT model.
Type: ARM_Cortex-M7.

FVP_MPS2_Cortex_M7.clk25Mhz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M7.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M7.default_ahb_slave

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_Cortex_M7.fvp_mps2

MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_Cortex_M7.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M7.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M7.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M7.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_Cortex_M7.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_Cortex_M7.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_Cortex_M7.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_Cortex_M7.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M7.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M7.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M7.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_Cortex_M7.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: MPS2_VGA.

FVP_MPS2_Cortex_M7.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.

Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M7.fvp_mps2.apb_ppc_iotss_expansion2`

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

`FVP_MPS2_Cortex_M7.fvp_mps2.clock50Hz`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.clockdivider`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.cmsdk_sysctrl`

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

`FVP_MPS2_Cortex_M7.fvp_mps2.cmsdk_watchdog`

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

`FVP_MPS2_Cortex_M7.fvp_mps2.cpu_wait_or_gate_0`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M7.fvp_mps2.cpu_wait_or_gate_1`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M7.fvp_mps2.dbgen_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_Cortex_M7.fvp_mps2.dma0`

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_Cortex_M7.fvp_mps2.dma0.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: ClockTimerThread.

FVP_MPS2_Cortex_M7.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M7.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M7.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M7.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M7.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M7.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M7.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M7.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M7.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M7.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M7.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M7.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_Cortex_M7.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_Cortex_M7.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_Cortex_M7.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M7.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_Cortex_M7.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_Cortex_M7.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_Cortex_M7.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_Cortex_M7.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_Cortex_M7.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_Cortex_M7.fvp_mps2.dma3_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_Cortex_M7.fvp_mps2.dma3_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_Cortex_M7.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M7.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_Cortex_M7.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M7.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_Cortex_M7.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.

Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M7.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M7.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_Cortex_M7.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.gpio_0_or_2

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M7.fvp_mps2.gpio_1_or_3

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M7.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_Cortex_M7.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: MPS2_LCD.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_Cortex_M7.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M7.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M7.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_Cortex_M7.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_Cortex_M7.fvp_mps2.spiden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M7.fvp_mps2.spniden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200
SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.acg_sram3
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.apb_ppc_iotss_subsystem0
IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.apb_ppc_iotss_subsystem1
IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.bus_error_warning_memory
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.clock32kHz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cpu0dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cpu1core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.cpu1dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.crypto_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.iotss_internal_sram2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.iotss_internal_sram3
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.iotss_systemcontrol
IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.iotss_systeminfo
IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mem_switch_internal_sram
Allow transactions to be routed arbitrarily.

Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_Cortex_M7.fvp_mps2.sse200.signal_router

Signal router.

Type: `SignalRouter`.

`FVP_MPS2_Cortex_M7.fvp_mps2.sse200.sys_ppu`

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_Cortex_M7.fvp_mps2.sse200.timer0`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M7.fvp_mps2.sse200.timer0.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.sse200.timer0.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.sse200.timer1`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_Cortex_M7.fvp_mps2.sse200.timer1.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.sse200.timer1.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.ssram1`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_Cortex_M7.fvp_mps2.ssram2`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_Cortex_M7.fvp_mps2.stub0`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_Cortex_M7.fvp_mps2.stub1`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_Cortex_M7.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer2`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer3`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_Cortex_M7.fvp_mps2.switch_PSRAM_M7`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M7.fvp_mps2.switch_svos_dualtimer`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_Cortex_M7.fvp_mps2.telnetterminal0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M7.fvp_mps2.telnetterminal1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M7.fvp_mps2.telnetterminal2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_Cortex_M7.fvp_mps2.touchscreen_interface`
MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

`FVP_MPS2_Cortex_M7.fvp_mps2.uart_overflows_or_gate`
Or Gate.
Type: `OrGate`.

10.11 FVP_MPS2_SSE-200_AEMv8M_pipeline

FVP_MPS2_SSE-200_AEMv8M_pipeline contains the following instances:

FVP_MPS2_SSE-200_AEMv8M_pipeline instances

FVP_MPS2_SSE_200_AEMv8M_pipeline

Type: `FVP_MPS2_SSE_200_AEMv8M_pipeline`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.aborting_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.clk25Mhz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.cpu0

ARM AEMv8M CT model.
Type: `ARM_AEMv8M`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.cpu1

ARM AEMv8M CT model.
Type: `ARM_AEMv8M`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2

MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.VGA_interface
VGA display interface between main bus and visualisation.
Type: `MPS2_VGA`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.ahb_ppc_iotss_expansion0
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.ahb_ppc_iotss_expansion1
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.apb_ppc_iotss_expansion0
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.apb_ppc_iotss_expansion1
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.apb_ppc_iotss_expansion2
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.clock50Hz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.cmsdk_sysctrl
Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.cmsdk_watchdog
ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.cpu_wait_or_gate_0
Or Gate.
Type: `OrGate`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.cpu_wait_or_gate_1
Or Gate.
Type: `OrGate`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dbgen_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma1_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma1_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma2_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma2_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma3_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.dma3_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.exclusive_monitor_psram
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.exclusive_monitor_psram_iotss
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.exclusive_monitor_zbtsram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.exclusive_monitor_zbtsram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.expansion_warning_memory
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.fpga_sysctrl
FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.fpga_sysctrl.callBack100HzCounter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.fpga_sysctrl.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.gpio_0_or_2
Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.gpio_1_or_3
Or Gate.

Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_cmsdk_dualtimer.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_cmsdk_dualtimer.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_exclusive_monitor_zbtsram1`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_exclusive_monitor_zbtsram2`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_lcd`
MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_mpc_iotss_ssram1`
IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event`
A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_secure_control_register_block`
MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_timer0`
ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_timer0.clk_div`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.platform_switch_dma1
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.signal_router
Signal router.
Type: SignalRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.smsc_91c111
SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.spiden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.spniden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200
SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.acg_sram3
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.apb_ppc_iotss_subsystem0
IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cpu0core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.iotss_internal_sram1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_e

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_e

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_e

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_e

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.signal_router

Signal router.
Type: `SignalRouter`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.sse200.timer1.counter`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.ssram1`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.ssram2`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.stub0`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.stub1`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.stub_i2c1`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.stub_i2s`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.stub_spi0`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.stub_spi2`
RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer`
Type: `SVOS_DualTimer`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer0`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.switch_PSRAM_M7

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.switch_svos_dualtimer

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.telnetterminal0

Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.telnetterminal1
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: `TelnetTerminal`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

FVP_MPS2_SSE_200_AEMv8M_pipeline.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: `OrGate`.

10.12 FVP_MPS2_SSE-200_Cortex-M33

FVP_MPS2_SSE-200_Cortex-M33 contains the following instances:

FVP_MPS2_SSE-200_Cortex-M33 instances

FVP_MPS2_SSE_200_Cortex_M33
Type: `FVP_MPS2_SSE_200_Cortex_M33`.

FVP_MPS2_SSE_200_Cortex_M33.clk25Mhz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.clk25khz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.cpu0
ARM CORTEXM33 CT model.
Type: `ARM_Cortex-M33`.

FVP_MPS2_SSE_200_Cortex_M33.cpu1
ARM CORTEXM33 CT model.
Type: `ARM_Cortex-M33`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2
MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: MPS2_VGA.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.

Type: `CMSDK_SysCtrl`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.cmsdk_watchdog`

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.cpu_wait_or_gate_0`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.cpu_wait_or_gate_1`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dbgen_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma0`

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma0.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma0_idau_labeller`

Type: `LabellerIdauSecurity`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma0_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).

Type: `PL080_DMAC`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: `ClockTimerThread`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).

Type: `ClockTimerThread64`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.

Type: `SchedulerThread`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.

Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma1_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma1_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).

Type: `PL080_DMAC`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus

transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.gpio_0_or_2

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.gpio_1_or_3

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: MPS2_LCD.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).

Type: `PL022_SSP_MPS2`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.pl022_ssp_mps2.prescaler`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.platform_bus_switch`

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.platform_switch_dma0`

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.platform_switch_dma1`

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.signal_router`

Signal router.
Type: `SignalRouter`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.smsc_91c111`

SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.spiden_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.spniden_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200`

SSE-200 subsystem.
Type: `SSE200`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.acg_cpu0`

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.acg_cpu1`

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.acg_sram0`

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.acg_sram1

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.acg_sram2

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cmsdk_dualtimer.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cordio_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cpu0core_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cpu0dbg_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cpu1core_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.cpu1dbg_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.crypto_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.dbg_ppu`
ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2`
Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

`FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3`
Global exclusive monitor.

Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.iotss_internal_sram3
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.iotss_systemcontrol
IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.iotss_systeminfo
IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mem_switch_internal_sram
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mem_switch_internal_sram_mpc
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mem_switch_ppu
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mhu0
IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.ram1_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.ram2_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.ram3_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.s32k_timer
ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.s32k_timer.clk_div
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.s32k_timer.counter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.s32k_watchdog
ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.secure_control_register_block
MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.secure_watchdog
ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.signal_router
Signal router.
Type: SignalRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.sys_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.timer0
ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.switch_PSRAM_M7
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.switch_svos_dualtimer
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.telnetterminal0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.telnetterminal1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: MPS2_TouchScreen.

FVP_MPS2_SSE_200_Cortex_M33.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: OrGate.

10.13 FVP_MPS2_SSE-200_Cortex-M55

FVP_MPS2_SSE-200_Cortex-M55 contains the following instances:

FVP_MPS2_SSE-200_Cortex-M55 instances

FVP_MPS2_SSE_200_Cortex_M55
Type: FVP_MPS2_SSE_200_Cortex_M55.

FVP_MPS2_SSE_200_Cortex_M55.clk25Mhz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.cpu0

ARM Cortex-M55 CT model.
Type: `ARM_Cortex-M55`.

FVP_MPS2_SSE_200_Cortex_M55.cpu1

ARM Cortex-M55 CT model.
Type: `ARM_Cortex-M55`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2

MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: `MPS2_VGA`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.cmsdk_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.cpu_wait_or_gate_0

Or Gate.
Type: `OrGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.cpu_wait_or_gate_1

Or Gate.
Type: `OrGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dbgen_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma0_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma0_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma3_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.dma3_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.

Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.gpio_0_or_2

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.gpio_1_or_3

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: HostBridge.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: MPS2_LCD.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.smsc_91c111

SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.spiden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.spniden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200

SSE-200 subsystem.
Type: `SSE200`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.acg_cpu0

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.acg_cpu1

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.acg_sram0

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.acg_sram1

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.acg_sram2

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cpu0dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cpu1core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.cpu1dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.crypto_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.dbg_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram2
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.iotss_internal_sram3
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.iotss_systemcontrol
IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.iotss_systeminfo
IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mem_switch_internal_sram
Allow transactions to be routed arbitrarily.

Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.signal_router

Signal router.

Type: `SignalRouter`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.sys_ppu`

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.timer0`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.timer0.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.timer0.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.timer1`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.timer1.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.sse200.timer1.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.ssram1`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.ssram2`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.stub0`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.stub1`

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer

Type: SVOS_DualTimer.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.switch_PSRAM_M7`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.switch_svos_dualtimer`
Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.telnetterminal0`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.telnetterminal1`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.telnetterminal2`
Telnet terminal interface.
Type: `TelnetTerminal`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.touchscreen_interface`
MPS2 Touch Screen.
Type: `MPS2_TouchScreen`.

`FVP_MPS2_SSE_200_Cortex_M55.fvp_mps2.uart_overflows_or_gate`
Or Gate.
Type: `OrGate`.

10.14 FVP_MPS2_SecurCore-SC000

FVP_MPS2_SecurCore-SC000 contains the following instances:

FVP_MPS2_SecurCore-SC000 instances

FVP_MPS2_SecurCore_SC000

Type: `FVP_MPS2_SecurCore_SC000`.

FVP_MPS2_SecurCore_SC000.armsc000ct

ARM SC000 CT model.
Type: `ARM_SC000`.

FVP_MPS2_SecurCore_SC000.clk25Mhz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.clk25khz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.dap_buslogger

Bus Logger.
Type: `PVBusLogger`.

FVP_MPS2_SecurCore_SC000.fvp_mps2

MPS2 DUT.
Type: `FVP_MPS2`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.GPIO0

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.GPIO_connection_test

Type: GPIO_Connection_Test.

FVP_MPS2_SecurCore_SC000.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: GPIO_Port_Transfer.

FVP_MPS2_SecurCore_SC000.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: GPIO1_Connection_Test.

FVP_MPS2_SecurCore_SC000.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SecurCore_SC000.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SecurCore_SC000.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_SecurCore_SC000.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_SecurCore_SC000.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.UART2

ARM CMSDK UART Module.
Type: CMSDK_UART.

FVP_MPS2_SecurCore_SC000.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: MPS2_VGA.

FVP_MPS2_SecurCore_SC000.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.
Type: `CMSDK_SysCtrl`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.cmsdk_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.cpu_wait_or_gate_0

Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.cpu_wait_or_gate_1

Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dbgen_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma0

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma0.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma0.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma0.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma0.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma0_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma0_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma1_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma1_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma2_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma2_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SecurCore_SC000.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SecurCore_SC000.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.exclusive_monitor_psram_iotss
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.exclusive_monitor_zbtsram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.exclusive_monitor_zbtsram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SecurCore_SC000.fvp_mps2.expansion_warning_memory
Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_SecurCore_SC000.fvp_mps2.fpga_sysctrl
FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_SecurCore_SC000.fvp_mps2.fpga_sysctrl.callBack100HzCounter
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.fpga_sysctrl.clockdivider
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.gpio_0_or_2
Or Gate.
Type: OrGate.

FVP_MPS2_SecurCore_SC000.fvp_mps2.gpio_1_or_3
Or Gate.
Type: OrGate.

FVP_MPS2_SecurCore_SC000.fvp_mps2.hostbridge
Host Socket Interface Component.

Type: HostBridge.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_audio

MPS2 Audio.
Type: MPS2_Audio.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_timer0

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_timer0.counter

Internal component used by SP804 Timer module.

Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_timer1.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.mps2_visualisation

Display window for MPS2 using Visualisation library.
Type: MPS2_Visualisation.

FVP_MPS2_SecurCore_SC000.fvp_mps2.niden_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SecurCore_SC000.fvp_mps2.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SecurCore_SC000.fvp_mps2.pl022_ssp_mps2

ARM PrimeCell Synchronous Serial Port(PL022).
Type: PL022_SSP_MPS2.

FVP_MPS2_SecurCore_SC000.fvp_mps2.pl022_ssp_mps2.prescaler

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.platform_bus_switch

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.platform_switch_dma0

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.platform_switch_dma1

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.signal_router
Signal router.
Type: `SignalRouter`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.smsc_91c111
SMSC 91C111 ethernet controller.
Type: `SMSC_91C111`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.spiden_or_gate
Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.spniden_or_gate
Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200
SSE-200 subsystem.
Type: `SSE200`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.acg_sram3
IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.apb_ppc_iotss_subsystem0
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.apb_ppc_iotss_subsystem1
IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cordio_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cpu0core_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.idau_labeller
Type: LabellerIdauSecurity.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.iotss_cpuidentity
IoT Subsystem CPU_IDENTITY registers.
Type: IoTSS_CPUIdentity.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.iotss_internal_sram0
RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.iotss_internal_sram1
RAM device, can be dynamic or static ram.

Type: RAMDevice.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: RAMDevice.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: IoTSS_SystemControl.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: IoTSS_SystemInfo.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: IoTSS_MessageHandlingUnit.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.

Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: IoTSS_MemoryProtectionController.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: OrGate.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: MPS2_SecureCtrl.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: CMSDK_Watchdog.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.signal_router

Signal router.
Type: SignalRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: CMSDK_Timer.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer

Type: `SVOS_DualTimer`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer1`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1`
Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer2`
ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1`
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new

ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer3
ARM Dual-Timer Module.
Type: CMSDK_DualTimer.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC000.fvp_mps2.switch_PSRAM_M7
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.switch_svos_dualtimer
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC000.fvp_mps2.telnetterminal0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SecurCore_SC000.fvp_mps2.telnetterminal1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SecurCore_SC000.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SecurCore_SC000.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: MPS2_TouchScreen.

FVP_MPS2_SecurCore_SC000.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: OrGate.

10.15 FVP_MPS2_SecurCore-SC300

FVP_MPS2_SecurCore-SC300 contains the following instances:

FVP_MPS2_SecurCore-SC300 instances

FVP_MPS2_SecurCore_SC300
Type: FVP_MPS2_SecurCore_SC300.

FVP_MPS2_SecurCore_SC300.armsc300ct
ARM SC300 CT model.
Type: ARM_SC300.

FVP_MPS2_SecurCore_SC300.clk25Mhz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC300.clk25khz
A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC300.dap_buslogger
Bus Logger.
Type: PVBusLogger.

FVP_MPS2_SecurCore_SC300.fvp_mps2
MPS2 DUT.
Type: FVP_MPS2.

FVP_MPS2_SecurCore_SC300.fvp_mps2.GPIO0
ARM PrimeCell General Purpose Input/Output.
Type: CMSDK_GPIO.

FVP_MPS2_SecurCore_SC300.fvp_mps2.GPIO1

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.GPIO2

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.GPIO3

ARM PrimeCell General Purpose Input/Output.
Type: `CMSDK_GPIO`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.GPIO_connection_test

Type: `GPIO_Connection_Test`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.GPIO_connection_test.GPIO0_port_trans

Type: `GPIO_Port_Transfer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.GPIO_connection_test.GPIO1_port_test

Type: `GPIO1_Connection_Test`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.PSRAM

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.PSRAM_M7

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.UART0

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.UART0.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.UART1

ARM CMSDK UART Module.
Type: `CMSDK_UART`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.UART1.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.UART2

ARM CMSDK UART Module.

Type: CMSDK_UART.

FVP_MPS2_SecurCore_SC300.fvp_mps2.UART2.clk_divider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC300.fvp_mps2.VGA_interface

VGA display interface between main bus and visualisation.
Type: MPS2_VGA.

FVP_MPS2_SecurCore_SC300.fvp_mps2.ahb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SecurCore_SC300.fvp_mps2.ahb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SecurCore_SC300.fvp_mps2.apb_ppc_iotss_expansion0

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SecurCore_SC300.fvp_mps2.apb_ppc_iotss_expansion1

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SecurCore_SC300.fvp_mps2.apb_ppc_iotss_expansion2

IoT Subsystem Peripheral Protection Controller.
Type: IoTSS_PeripheralProtectionController.

FVP_MPS2_SecurCore_SC300.fvp_mps2.clock50Hz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC300.fvp_mps2.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: ClockDivider.

FVP_MPS2_SecurCore_SC300.fvp_mps2.cmsdk_sysctrl

Cortex-M Simple System Control.
Type: CMSDK_SysCtrl.

FVP_MPS2_SecurCore_SC300.fvp_mps2.cmsdk_watchdog

ARM Watchdog Module.

Type: `CMSDK_Watchdog`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.cpu_wait_or_gate_0`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.cpu_wait_or_gate_1`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dbgen_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dma0`

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dma0.timer`

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dma0.timer.timer`

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dma0.timer.timer.thread`

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dma0.timer.timer.thread_event`

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dma0_idau_labeller`

Type: `LabellerIdauSecurity`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.dma0_securitymodifier`

Type: `SecurityModifier`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma1

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma1.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma1.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread64.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma1.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: SchedulerThread.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma1.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma1_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma1_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma2

ARM PrimeCell DMA Controller(PL080/081).
Type: PL080_DMAC.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma2.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: ClockTimerThread.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma2.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma2.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma2.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma2_idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma2_securitymodifier

Type: `SecurityModifier`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma3

ARM PrimeCell DMA Controller(PL080/081).
Type: `PL080_DMAC`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma3.timer

A ClockTimerThread(64) is a drop-in replacement for a ClockTimer(64) component. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma3.timer.timer

A ClockTimerThread64 is a drop-in replacement for ClockTimer64. The main difference to the ClockTimer component is that the ClockTimerThread runs the signal() callback from a proper scheduler thread. This mean that the signal() function may directly or indirectly invoke wait() functions to wait for time or events. This is not allowed for the ClockTimer component which does not use a thread. Components which issue bus transactions from within the timer signal() callback must use ClockTimerThread(64) rather than ClockTimer(64).
Type: `ClockTimerThread64`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma3.timer.timer.thread

A SchedulerThread instance represents a co-routine thread in the simulation.
Type: `SchedulerThread`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma3.timer.timer.thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: SchedulerThreadEvent.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma3_idau_labeller

Type: LabellerIdauSecurity.

FVP_MPS2_SecurCore_SC300.fvp_mps2.dma3_securitymodifier

Type: SecurityModifier.

FVP_MPS2_SecurCore_SC300.fvp_mps2.exclusive_monitor_psram

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.exclusive_monitor_psram_iotss

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.exclusive_monitor_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC300.fvp_mps2.exclusive_monitor_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC300.fvp_mps2.exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.exclusive_squasher

Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SecurCore_SC300.fvp_mps2.expansion_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: WarningMemory.

FVP_MPS2_SecurCore_SC300.fvp_mps2.fpga_sysctrl

FPGA SysCtrl timers LEDs and switches.
Type: FPGA_SysCtrl.

FVP_MPS2_SecurCore_SC300.fvp_mps2.fpga_sysctrl.callBack100HzCounter

Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC300.fvp_mps2.fpga_sysctrl.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.gpio_0_or_2

Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.gpio_1_or_3

Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.hostbridge

Host Socket Interface Component.
Type: `HostBridge`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mem_switch_extra_psram_iotss

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mem_switch_extra_psram_mps2

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mpc_iotss_ssram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mpc_iotss_ssram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mpc_iotss_ssram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mpc_iotss_ssram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_audio

MPS2 Audio.
Type: `MPS2_Audio`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_cmsdk_dualtimer.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_exclusive_monitor_zbtsram1

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_exclusive_monitor_zbtsram2

Global exclusive monitor.
Type: `PVBusExclusiveMonitor`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_lcd

MPS2 LCD I2C interface.
Type: `MPS2_LCD`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_mpc_iotss_ssram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_mpc_iotss_ssram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_secure_control_register_block

MPS2 Secure Control Register Block.

Type: `MPS2_SecureCtrl`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_timer0`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_timer0.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_timer0.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_timer1`

ARM Timer Module.
Type: `CMSDK_Timer`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_timer1.clk_div`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_timer1.counter`

Internal component used by SP804 Timer module.
Type: `CounterModule`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.mps2_visualisation`

Display window for MPS2 using Visualisation library.
Type: `MPS2_Visualisation`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.niden_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.nmi_or_gate`

Or Gate.
Type: `OrGate`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.pl022_ssp_mps2`

ARM PrimeCell Synchronous Serial Port(PL022).
Type: `PL022_SSP_MPS2`.

`FVP_MPS2_SecurCore_SC300.fvp_mps2.pl022_ssp_mps2.prescaler`

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.

Type: ClockDivider.

FVP_MPS2_SecurCore_SC300.fvp_mps2.platform_bus_switch
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC300.fvp_mps2.platform_switch_dma0
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC300.fvp_mps2.platform_switch_dma1
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC300.fvp_mps2.signal_router
Signal router.
Type: SignalRouter.

FVP_MPS2_SecurCore_SC300.fvp_mps2.smsc_91c111
SMSC 91C111 ethernet controller.
Type: SMSC_91C111.

FVP_MPS2_SecurCore_SC300.fvp_mps2.spiden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_SecurCore_SC300.fvp_mps2.spniden_or_gate
Or Gate.
Type: OrGate.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200
SSE-200 subsystem.
Type: SSE200.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.acg_cpu0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.acg_cpu1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.acg_sram0
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.acg_sram1
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.acg_sram2
IoT Subsystem Access Control Gate.
Type: IoTSS_AccessControlGate.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.acg_sram3

IoT Subsystem Access Control Gate.
Type: `IoTSS_AccessControlGate`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.apb_ppc_iotss_subsystem0

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.apb_ppc_iotss_subsystem1

IoT Subsystem Peripheral Protection Controller.
Type: `IoTSS_PeripheralProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.bus_error_warning_memory

Memory that prints warnings, and RAZ/WIs or aborts.
Type: `WarningMemory`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.clock32kHz

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.clockdivider

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cmsdk_dualtimer

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cmsdk_dualtimer.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cmsdk_dualtimer.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cmsdk_dualtimer.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cmsdk_dualtimer.counter1
Internal component used by SP804 Timer module.
Type: CounterModule.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cordio_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cpu0core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cpu0dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cpu1core_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.cpu1dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.crypto_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.dbg_ppu
ARM Power Policy Unit (PPU) architectural model.
Type: PPUv0.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram0
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram1
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram2
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.exclusive_monitor_iotss_internal_sram3
Global exclusive monitor.
Type: PVBusExclusiveMonitor.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.exclusive_squasher
Squashes the exclusive attribute on bus transactions.
Type: PVBusExclusiveSquasher.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.idau_labeller

Type: `LabellerIdauSecurity`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.iotss_cpuidentity

IoT Subsystem CPU_IDENTITY registers.
Type: `IoTSS_CPUIdentity`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.iotss_internal_sram0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.iotss_internal_sram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.iotss_internal_sram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.iotss_internal_sram3

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.iotss_systemcontrol

IoT Subsystem System Control registers.
Type: `IoTSS_SystemControl`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.iotss_systeminfo

IoT Subsystem System Information registers.
Type: `IoTSS_SystemInfo`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mem_switch_internal_sram

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mem_switch_internal_sram_mpc

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mem_switch_ppu

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mhu0

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mhu1

IoT Subsystem Message Handling Unit.
Type: `IoTSS_MessageHandlingUnit`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram0

IoT Subsystem Memory Protection Controller.

Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram0.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram1

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram1.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram2

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram2.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram3

IoT Subsystem Memory Protection Controller.
Type: `IoTSS_MemoryProtectionController`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.mpc_iotss_internal_sram3.gating_disabled_thread_event

A SchedulerThreadEvent instance is an auto-reset boolean condition variable other threads can wait on.
Type: `SchedulerThreadEvent`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.nmi_or_gate

Or Gate.
Type: `OrGate`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.nonsecure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.ram0_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.ram1_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.ram2_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.ram3_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.s32k_timer

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.s32k_timer.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.s32k_timer.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.s32k_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.secure_control_register_block

MPS2 Secure Control Register Block.
Type: `MPS2_SecureCtrl`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.secure_watchdog

ARM Watchdog Module.
Type: `CMSDK_Watchdog`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.signal_router

Signal router.
Type: `SignalRouter`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.sys_ppu

ARM Power Policy Unit (PPU) architectural model.
Type: `PPUv0`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.timer0

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.timer0.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.timer0.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.timer1

ARM Timer Module.
Type: `CMSDK_Timer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.timer1.clk_div

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.sse200.timer1.counter

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.ssram1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.ssram2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.stub0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.stub1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.stub_i2c1

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.stub_i2s

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.stub_spi0

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.stub_spi2

RAM device, can be dynamic or static ram.
Type: `RAMDevice`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer

Type: `SVOS_DualTimer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer0

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer0.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer0.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer1

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer1.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer1.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer2

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer2.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer2.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer3

ARM Dual-Timer Module.
Type: `CMSDK_DualTimer`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div0

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer3.clk_div1

A ClockDivider is a library component that takes a ClockSignal on its input port (which could come from the output of a MasterClock, or from another ClockDivider), and generates a new ClockSignal on its output port, representing a clock frequency that is related to the input clock by the ratio of the multiply and divide parameters.
Type: `ClockDivider`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter0

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.svos_dualtimer.svos_dualtimer3.counter1

Internal component used by SP804 Timer module.
Type: `CounterModule`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.switch_PSRAM_M7

Allow transactions to be routed arbitrarily.
Type: `PVBusRouter`.

FVP_MPS2_SecurCore_SC300.fvp_mps2.switch_svos_dualtimer
Allow transactions to be routed arbitrarily.
Type: PVBusRouter.

FVP_MPS2_SecurCore_SC300.fvp_mps2.telnetterminal0
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SecurCore_SC300.fvp_mps2.telnetterminal1
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SecurCore_SC300.fvp_mps2.telnetterminal2
Telnet terminal interface.
Type: TelnetTerminal.

FVP_MPS2_SecurCore_SC300.fvp_mps2.touchscreen_interface
MPS2 Touch Screen.
Type: MPS2_TouchScreen.

FVP_MPS2_SecurCore_SC300.fvp_mps2.uart_overflows_or_gate
Or Gate.
Type: OrGate.