Arm® MPS3 FPGA Prototyping Board

Technical Reference Manual

arm

100765_0000_04_en

Arm® MPS3 FPGA Prototyping Board

Technical Reference Manual

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	01 July 2017	Non-Confidential	First issue of TRM
0000-01	16 October 2017	Non-Confidential	Second issue of TRM
0000-02	27 April 2018	Non-Confidential	Third issue of TRM
0000-03	01 August 2019	Non-Confidential	Fourth issue of TRM
0000-04	29 May 2020	Non-Confidential	Fifth issue of TRM.

Non-Confidential Proprietary Notice

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

www.arm.com

Conformance Notices

This section contains conformance notices.

Federal Communications Commission Notice

This device is test equipment and consequently is exempt from part 15 of the FCC Rules under section 15.103 (c).

CE Conformity

The *Waste Electrical and Electronic Equipment* (WEEE) marking, that is, the crossed out wheelie-bin figure, indicates that this product must not be disposed of with general waste within the European Union. To prevent possible harm to the environment from uncontrolled waste disposal, the user is required to recycle the product responsibly to promote reuse of material resources. To comply with EU law, you must dispose of the product in one of the following ways:

- Return it to the distributer where it was purchased. The distributer is required to arrange free collection when requested.
- Recycle it using local WEEE recycling facilities. These facilities are now very common and might provide free collection.
- If purchased directly from Arm, Arm provides free collection. Please e-mail *weee@arm.com* for instructions.

The CE Declaration of Conformity for this product is available on request.

The system should be powered down when not in use.

It is recommended that ESD precautions be taken when handling this product.

The product generates, uses, and can radiate radio frequency energy and may cause harmful interference to radio communications. There is no guarantee that interference will not occur in a particular installation. If this equipment causes harmful interference to radio or television reception, which can be determined by turning the equipment off or on, you are encouraged to try to correct the interference by one or more of the following measures:

- Ensure attached cables do not lie across any sensitive equipment.
- Reorient the receiving antenna.
- Increase the distance between the equipment and the receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Note

It is recommended that wherever possible shielded interface cables be used.

Contents

Arm® MPS3 FPGA Prototyping Board Technical Reference Manual

Preface

Chapter 1 Introduction

Chapter 2 Hardware description

Preface

This preface introduces the *Arm® MPS3 FPGA Prototyping Board Technical Reference Manual*.

It contains the following:

- *About this book* on page 7.
- *Feedback* on page 10.

About this book

This book describes the Arm® MPS3 FPGA Prototyping Board.

Intended audience

This book is written for experienced hardware and software developers to enable them to perform FPGA development using the MPS3 board.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction

This chapter introduces the MPS3 board.

Chapter 2 Hardware description

This chapter describes the MPS3 board hardware.

Chapter 3 Configuration

This chapter describes the powerup and configuration processes of the MPS3 board.

Appendix A Signal descriptions

This appendix lists the signals at the interface connectors of the MPS3 board.

Appendix B Specifications

This appendix contains electrical specifications of the MPS3 board.

Appendix C Revisions

This appendix describes the technical changes between released issues of this book.

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.

Typographic conventions

italic

Introduces special terminology, denotes cross-references, and 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`

Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`

Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`

Denotes language keywords when used outside example code.

`<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.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

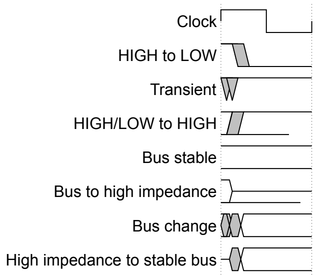


Figure 1 Key to timing diagram conventions

Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name, n denotes an active-LOW signal.

Additional reading

This book contains information that is specific to this product. See the following documents for other relevant information.

Arm publications

- *Arm® Cortex®-M System Design Kit Technical Reference Manual* (DDI 0479).
- *Application Note AN524 Example SSE-200 Subsystem for MPS3* (DAI 0524).
- *Application Note AN533 Blinky example FPGA image for the MPS3 Prototyping Board* (DAI 0533)
- *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual* (DDI 0571).
- *Arm® CoreLink™ SSE-200 Subsystem Technical Overview* (DTO 0051).
- *Arm® CoreLink™ SSE-100 Subsystem Technical Reference Manual* (DDI 0551).
- *Arm® DS-5 Arm DSTREAM User Guide* (100955).
- *Arm® DS-5 Using the Debug Hardware Configuration Utilities* (DUI 0498).
- *CoreSight™ Components Technical Reference Manual* (DDI 0314).
- *CoreSight™ Trace Memory Controller Technical Reference Manual* (DDI 0461).

Other publications

- See the Xilinx website *https://www.xilinx.com* for information about the Xilinx Kintex Ultrascale XCKU115-1FLVB1760C FPGA.

Feedback

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

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

- The title *Arm MPS3 FPGA Prototyping Board Technical Reference Manual*.
- The number 100765_0000_04_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.

Chapter 1
Introduction

This chapter introduces the MPS3 board.

It contains the following sections:

1.1 Precautions

You can take certain precautions to ensure safety and to prevent damage to your MPS3 board.

This section contains the following subsections:

- *1.1.1 Ensuring safety* on page 1-12.
- *1.1.2 Operating temperature* on page 1-12.
- *1.1.3 Preventing damage* on page 1-12.

1.1.1 Ensuring safety

An on-board connector supplies 12V DC to the MPS3 board.

——— **Warning** ———

Do not use the MPS3 board near equipment that is sensitive to electromagnetic emissions, for example, medical equipment.

———————————

1.1.2 Operating temperature

The MPS3 board has been tested in the temperature range 0°C to 40°C.

1.1.3 Preventing damage

The MPS3 board is intended for use within a laboratory or engineering development environment.

——— **Caution** ———

To avoid damage to the MPS3 board, observe the following precautions:

- Connect the external power supply to the board before starting the powerup process.
- Never subject the board to high electrostatic potentials. Observe *ElectroStatic Discharge* (ESD) precautions when handling any board.
- Always wear a grounding strap when handling the board.
- Only hold the board by the edges.
- Avoid touching the component pins or any other metallic element.
- Do not fit an Arduino Shield, Pmod expansion shield, FMC-HPC expansion board, or configuration microSD card while the MPS3 board is powered.

———————————

1.2 About the MPS3 board

The MPS3 board is an FPGA *Internet of Things* (IoT) development platform. The board is designed to support Arm Cortex-M class and small to medium Arm Cortex-A and Cortex-R class processors, or dedicated custom designs.

Major features of the MPS3 board

The MPS3 board contains a Xilinx Kintex Ultrascale XCKU115-1FLVB1760C FPGA, support logic, and peripheral interfaces that provide access to the FPGA and I/O interfaces.

The feature-rich set of user peripherals connects directly to the FPGA and provides flexibility for the user. The peripherals can be included in a custom design as required.

Two Arduino expansion Shield slots enable connection of sensors, motors, and other design-specific peripherals. The MPS3 board also provides expansion through four *Peripheral Module* (Pmod) expansion ports, and an *FPGA Mezzanine Card High Pin Count* (FMC-HPC) expansion port.

An on-board *Motherboard Configuration Controller* (MCC) controls the board and configures the FPGA in a way similar to other Arm development boards including boards in the Arm Versatile™ Express range.

Uses of the MPS3 board

The MPS3 board enables FPGA prototyping of complex designs:

- Software development:
  - Linux development on Cortex-A or Cortex-R class processors.
  - mbedOS, *Cortex Microcontroller Software Interface Standard* (CMSIS), *Real-Time Operating System* (RTOS) development on Cortex-M class processors.
  - Bare metal development.
- Tool development.

Major components and systems of the MPS3 board

The MPS3 board provides:

- One Kintex XCKU115 FPGA.
- Board interfaces:
  - Ethernet 10/100.
  - AC97 audio.
  - HDMI video up to 1080p.
  - Dual USB-A port.
- Expansion connectivity:
  - Two Arduino Shield interfaces for custom peripherals.
  - FMC-HPC expansion for up to 160 I/O, 10 multi-*GigaBit Transceivers* (GBT), and clocks.
  - Four Pmod interfaces.
- *Quarter Video Graphics Array* (QVGA) CLCD with touchscreen.
- Memory:
  - 4GB DDR4 with capacity for up to 8GB.
  - 8MB user *Quad Serial Peripheral Interface* (QSPI) flash for boot.
  - Up to 8MB of FPGA *Block RAM* (BRAM).
  - 16GB eMMC.
  - microSD card interface.
- Clocks:
  - *Real-Time Clock* (RTC).
  - Five programmable clocks.
  - One fixed 24MHz clock.
- User board components:
  - Ten user LEDs.
  - Eight user switches.
  - Two user push buttons.

- Reset push buttons and power indicator LEDs.
- Debug support:
  - — P-JTAG processor debug.
  - — F-JTAG (FPGA) debug.
  - — *Serial Wire Debug* (SWD).
  - — 16-bit trace.
  - — 4-bit trace.
  - — On-board CMSIS-DAP.
  - — Four serial ports over USB.

1.3 Location of components on the MPS3 board

The following figure shows the physical layout of the MPS3 board.

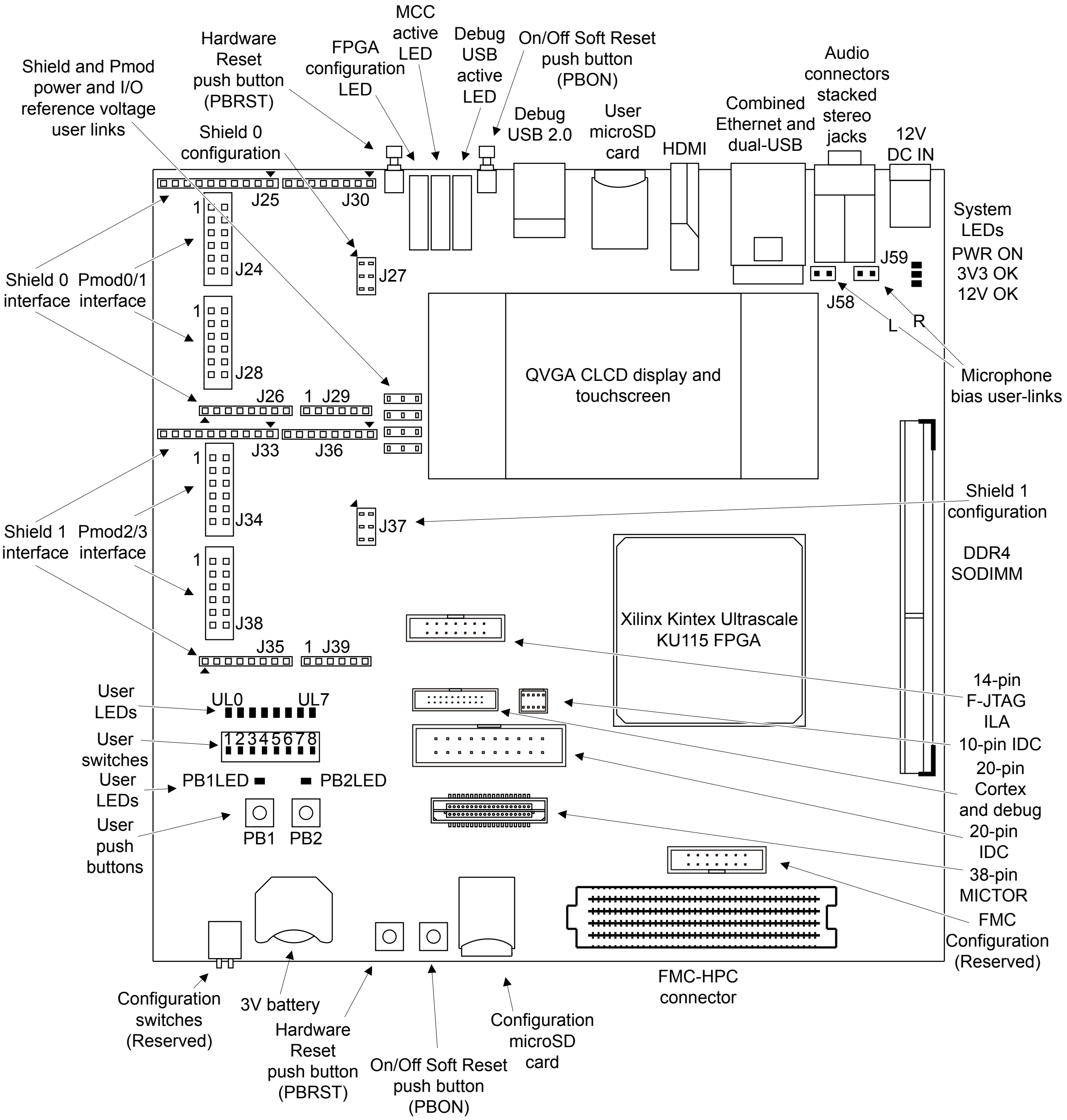


Figure 1-1 Layout of the MPS3 board

See *2.12 On-board user components* on page 2-39 for a description of the use of the Shield and Pmod power and I/O reference voltage user links.

Note

The configuration switches are reserved. For correct operation, you must ensure that both switches are in the OFF position, that is, pointing away from the board.

Chapter 2
Hardware description

This chapter describes the MPS3 board hardware.

It contains the following sections:

2.1 Overview of the board hardware

The MPS3 board provides access to the Kintex XCKU115 FPGA and peripherals to enable FPGA prototyping and software development.

The following figure shows the hardware infrastructure of the MPS3 board.

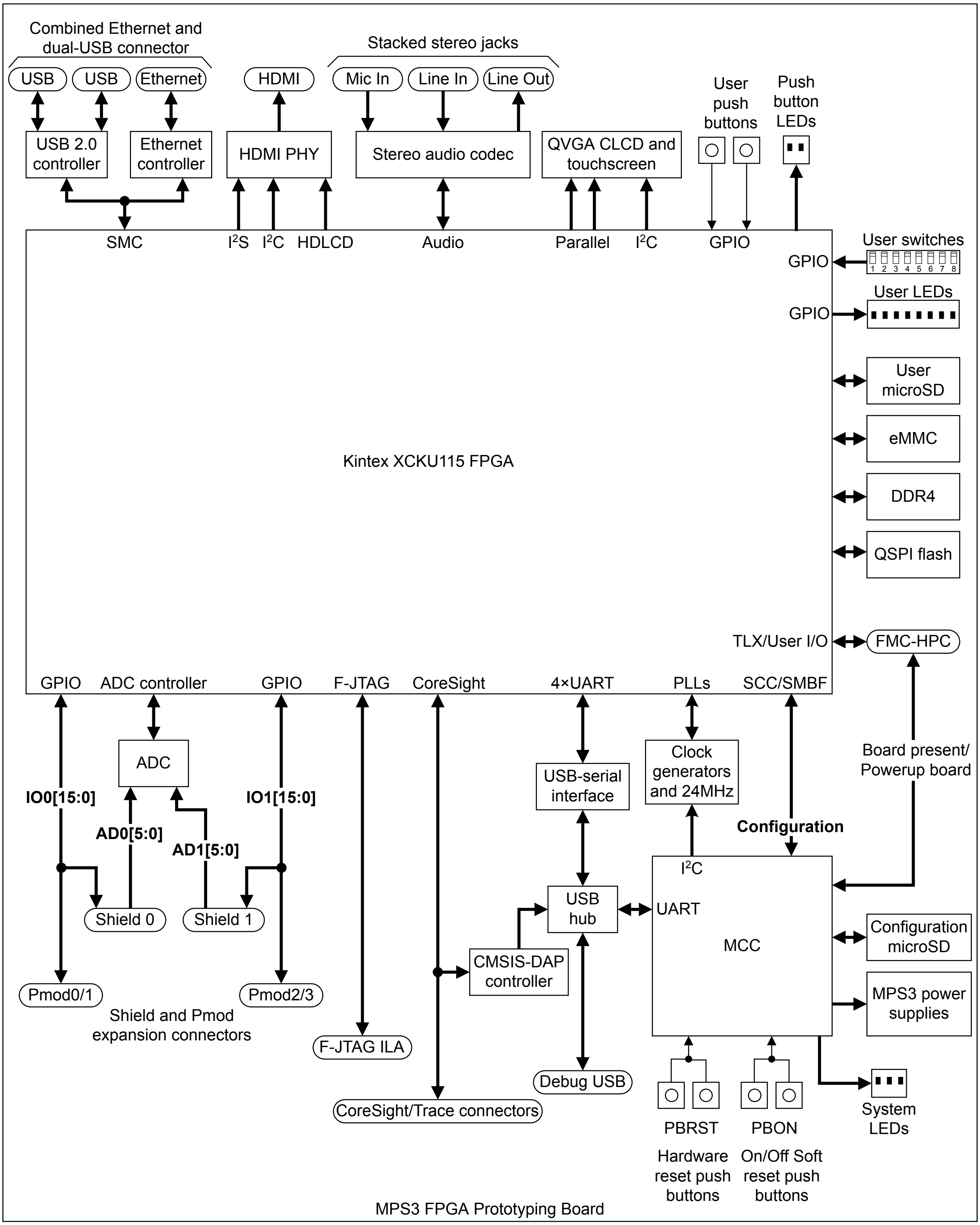


Figure 2-1 Hardware infrastructure of the MPS3 board

The MPS3 board contains the following components:

- One Xilinx Kintex Ultrascale XCKU115-1FLVB1760C FPGA:

  - 1451k logic cells.
  - Four UARTs (connected to a USB to serial hub for connection to the host computer).
  - Support for encrypted FPGA images and Partial Reconfiguration.
- User memory system:
  - Up to 8MB internal BRAM.
  - microSD card interface.
  - 8MB external QSPI flash.
  - 4GB, 64-bit external DDR4 SODIMM.
  - 16GB, 8-bit external eMMC.
- *Motherboard Configuration Controller* (MCC) that controls the MPS3 board, and supports board configuration at powerup or reset:
  - FPGA configuration.
  - Board configuration.
  - Supervises user update of the board configuration files in the configuration microSD card.
  - Two hardware reset buttons, both labeled *PBRST*, and two On/Off soft reset buttons, both labeled *PBON*.
- Ethernet 10/100 port and Ethernet controller that connects to the *Static Memory Controller* (SMC) interface in the FPGA.
- HDMI port and HDMI controller:
  - Inputs 24-bit RGB data from the HDLCD controller in the FPGA.
  - Configured over I^2C directly from FPGA.
  - Supports an I^2S audio connection to the FPGA.
  - Drives the HDMI connector.
- Audio codec:
  - Provides stereo Line In, stereo Line Out, and stereo Microphone Input In to the stacked stereo jack audio interface.
  - Configured over I^2C from the I^2C controller in the FPGA.
  - I^2S audio connection to the I^2S in the FPGA.
- Shield expansion:
  - Two Arduino expansion interfaces for Shields for custom peripherals or off-the-shelf sensor interfaces such as WiFi, Bluetooth, Proximity detectors, or Gyro sensors.
  - Each interface connects: 16 × digital 3V3 I/O or 16 × digital 5V I/O, voltage references, and six analog inputs from each Shield.
- *Peripheral Module* (Pmod) interface expansion:
  - Four Pmod expansion connectors that use the same I/O on an interface that is shared with the Shield connectors, Type 2A/3/4 support.
  - Requires Pmod adapters that provide 8 × analog 3V3 I/O, or 8 × analog 5V I/O.
- FMC-HPC expansion:
  - Connector to enable fitting of an *FPGA Mezzanine Card* (FMC) *High Pin Count* (HPC) expansion card to the MPS3 board.
  - A 14-pin FMC configuration connector and configuration EEPROM to enable configuration of Arm FMC boards.
- User switches, LEDs, and push buttons that connect directly to GPIO in the FPGA:
  - One 8-way user DIP switch.
  - Ten user LEDs.
  - Two user push buttons.
- System LEDs:
  - Green *12V OK* LED.
  - Orange *3V3 OK* LED.
  - Green *PWR ON* LED.
  - Green *FPGA configuration* LED.
  - Green *MCC active* LED.
  - Orange *Debug USB active* LED.

- — Green LED. Ethernet speed indicator-incorporated into combined Ethernet and dual-USB connector.
- — Yellow LED. Ethernet link and activity indicator-incorporated into combined Ethernet and dual-USB connector.

Related information

1.3 Location of components on the MPS3 board on page 1-15

2.2 Example Cortex®-M33 IoT Kit subsystem design

The MPS3 board is an FPGA *Internet of Things* (IoT) development platform. The board is designed to support Arm Cortex-M class and small to medium Arm Cortex-A and Cortex-R class processors, or dedicated customs designs.

The following figure shows an example Cortex-M33 IoT Kit subsystem design for the MPS3 board. You can implement this example design in MPS3. See *Application Note AN524 Example SSE-200 Subsystem for MPS3*.

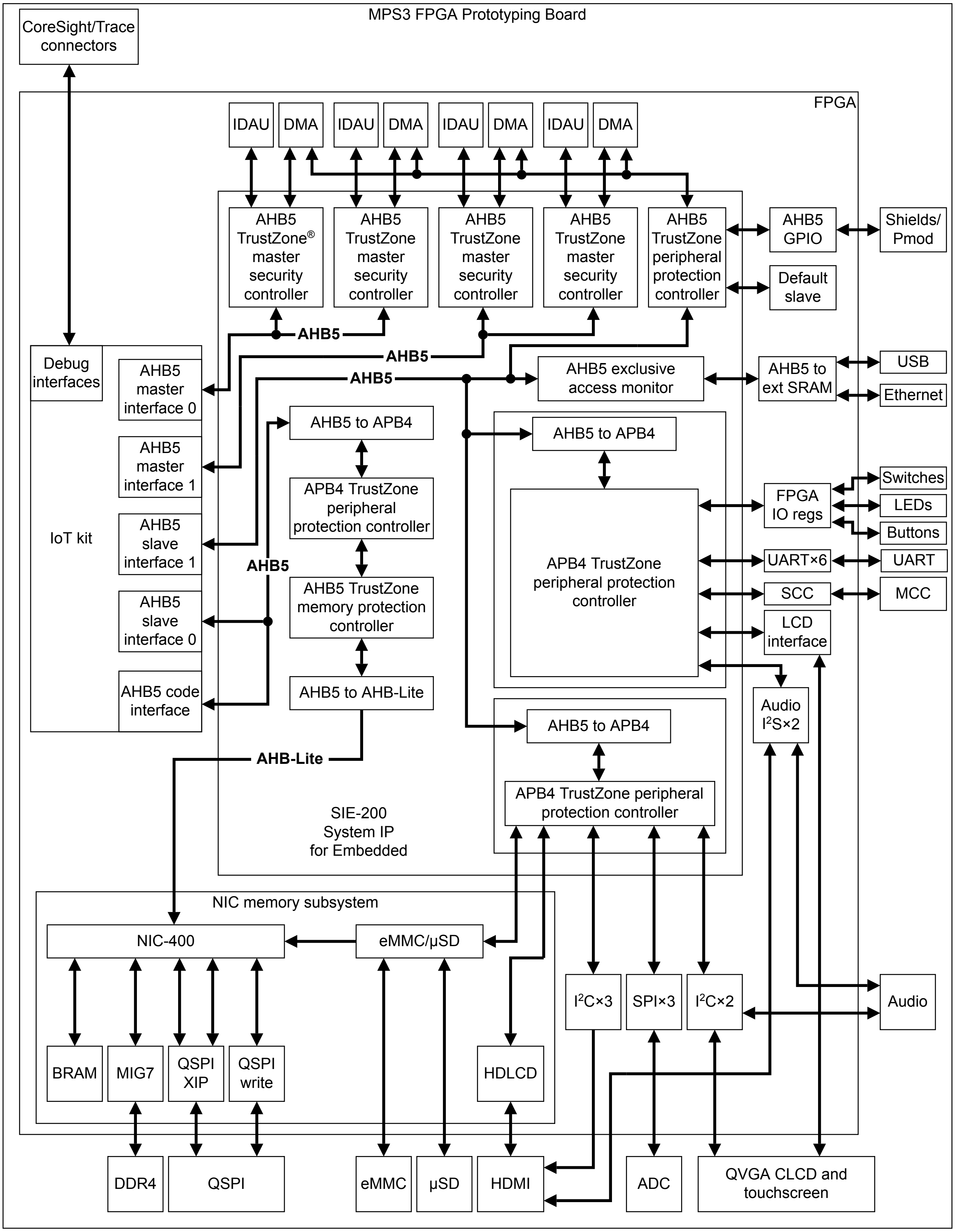

MPS3 FPGA Prototyping Board
CoreSight/Trace connectors
FPGA
IDAU
DMA
IDAU
DMA
IDAU
DMA
IDAU
DMA
AHB5 TrustZone® master security controller
AHB5 TrustZone master security controller
AHB5 TrustZone master security controller
AHB5 TrustZone master security controller
AHB5 TrustZone peripheral protection controller
AHB5 GPIO
Shields/ Pmod
Default slave
AHB5
AHB5
AHB5
Debug interfaces
AHB5 master interface 0
AHB5 master interface 1
AHB5 slave interface 1
AHB5 slave interface 0
AHB5 code interface
IoT kit
AHB5 exclusive access monitor
AHB5 to ext SRAM
USB
Ethernet
AHB5 to APB4
APB4 TrustZone peripheral protection controller
AHB5
AHB5 TrustZone memory protection controller
AHB5 to AHB-Lite
AHB5 to APB4
APB4 TrustZone peripheral protection controller
FPGA IO regs
Switches
LEDs
Buttons
UART×6
UART
SCC
MCC
LCD interface
Audio I²S×2
AHB5 to APB4
APB4 TrustZone peripheral protection controller
AHB-Lite
SIE-200 System IP for Embedded
NIC memory subsystem
NIC-400
eMMC/µSD
BRAM
MIG7
QSPI XIP
QSPI write
HDLCD
I²C×3
SPI×3
I²C×2
Audio
DDR4
QSPI
eMMC
µSD
HDMI
ADC
QVGA CLCD and touchscreen

Figure 2-2 Example IoT Kit subsystem design for MPS3

2.3 Clocks

The MPS3 board provides fixed and programmable clocks to drive the FPGA and board interfaces.

The following figure shows a functional overview of the clock systems of the MPS3 board. In this figure, the image implements OSC4 as the audio clock and OSC5 as the HDLCD clock.

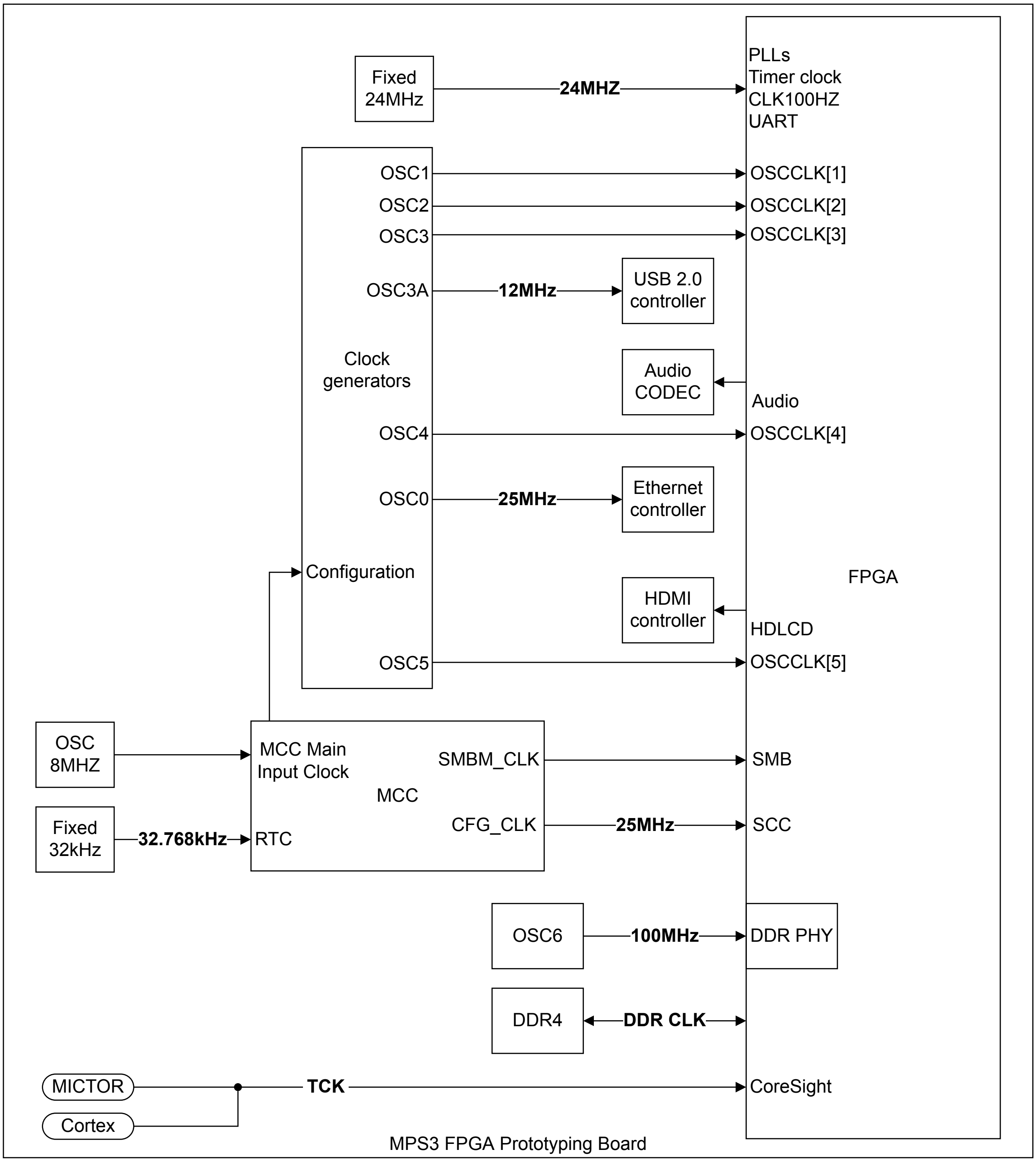


Figure 2-3 Overview of MPS3 board clocks

The *Motherboard Configuration Controller* (MCC) configures the programmable OSCs at powerup using the default values, which are defined in the MPS3 board configuration application note `.txt` file in the configuration microSD card. The system register interface can implement runtime control of the OSCs.

PLLs within the FPGA can use the reference 24MHz to generate other fixed internal frequencies.

The following table lists the MPS3 board clocks and their characteristics.

Table 2-1 MPS3 board clocks

Clock name: Source	Programmable frequency range	Destination: Description
24MHz	24MHz fixed	**FPGA OSCCLK[0]**
OSC1	2-230MHz	**FPGA OSCCLK[1]**
OSC2	2-230MHz	**FPGA OSCCLK[2]**
OSC3	2-230MHz	**FPGA OSCCLK[3]**
OSC3A	12MHz default	USB 2.0 controller. 12MHz is required by USB 2.0 controller.
OSC4	2-230MHz	**FPGA OSCCLK[4]**. The preferred use for this clock is as the clock for the audio codec and FGPA audio interface. In this case, this clock is the source clock for: • **AACI_MCLK**. • **AACI_SCLK**. • **AACI_LRCLK**.
OSC0	25MHz default	Ethernet controller. 25MHz required by Ethernet controller.
OSC5	2-230MHz	**FPGA OSCCLK[5]**. The preferred use for this clock is for the HDMI controller and HDLCD interface in FPGA.
OSC6	100MHz default	DDR PHY. 100MHz required by DDR PHY.
OSC8MHz	8MHz fixed	MCC main input clock.
OSC32K	32.768kHz fixed	Input for *Real Time Clock* (RTC) in MCC.
TCK: JTAG	-	Source clock for JTAG debug system. Frequency depends on the debugger setting.
CFGCLK: MCC	25MHz fixed	Configuration clock for FPGA.

Related information

1.3 Location of components on the MPS3 board on page 1-15

2.4 Reset, powerup, and configuration

The MPS3 board provides five external resets to the FPGA.

Overview of reset system

The MPS3 board provides a hardware reset, and a software reset.

There are two hardware reset buttons. They perform the same function and both are labeled *PBRST*. Pressing one of them puts the system into the standby state.

There are two On/Off soft reset buttons. Both are labeled *PBON*. Pressing one of them performs a software reset, or if the board is already in the standby state, powers up the system.

The following figure shows the MPS3 board reset system, where the FPGA contains a user image.

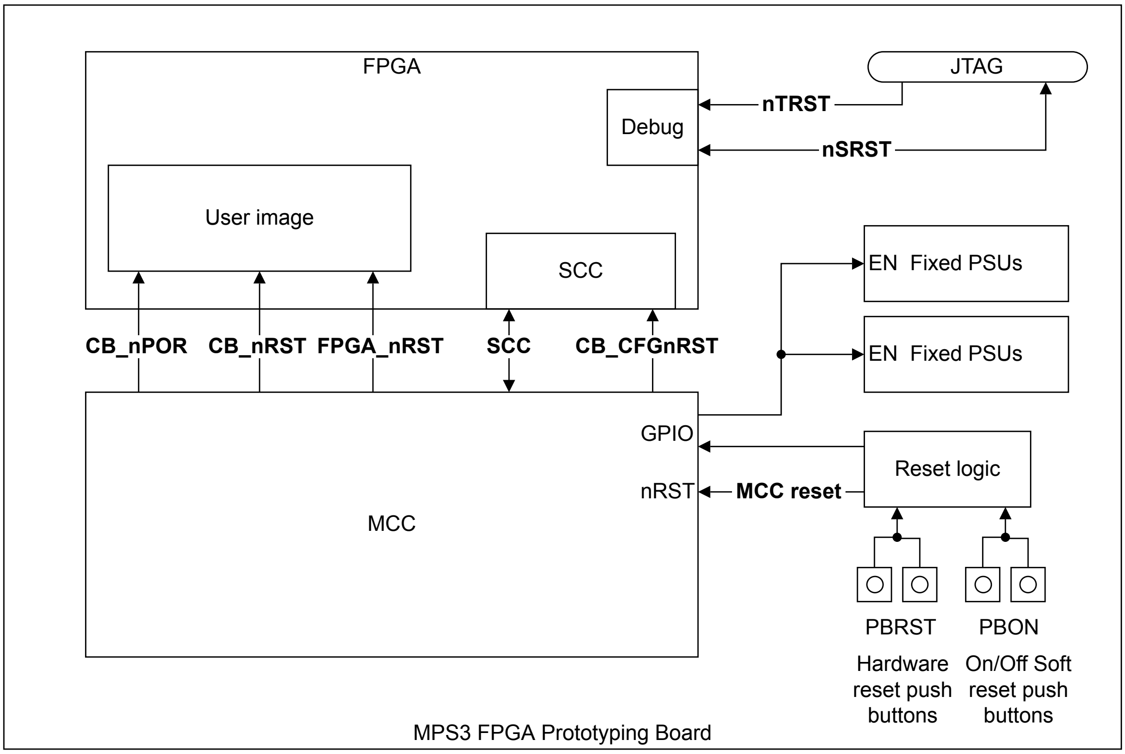


Figure 2-4 MPS3 board reset system

FPGA resets

FPGA_nRST
Board and FPGA reset including FPGA PLLs.

CB_nPOR
The main powerup reset for the FPGA image logic. If a *System Control Processor* (SCP) is present in the design, releasing this reset might also trigger the powerup reset sequencing.

nTRST
Resets the CoreSight DAP.

CB_nRST, nSRST
CB_nRST is the core reset. These inputs are ANDed together in the FPGA. They initiate operation of the processors and enable the debug tools to debug the processors before they leave reset.

CB_CFGnRST

The reset signal for the serial interface of the *Serial Configuration Controller* (SCC).

CPUWAIT

Core register that is used to release processor core or cores from reset.

Reset sequence

The following figure shows the MPS3 board reset and powerup timing cycle including board configuration.

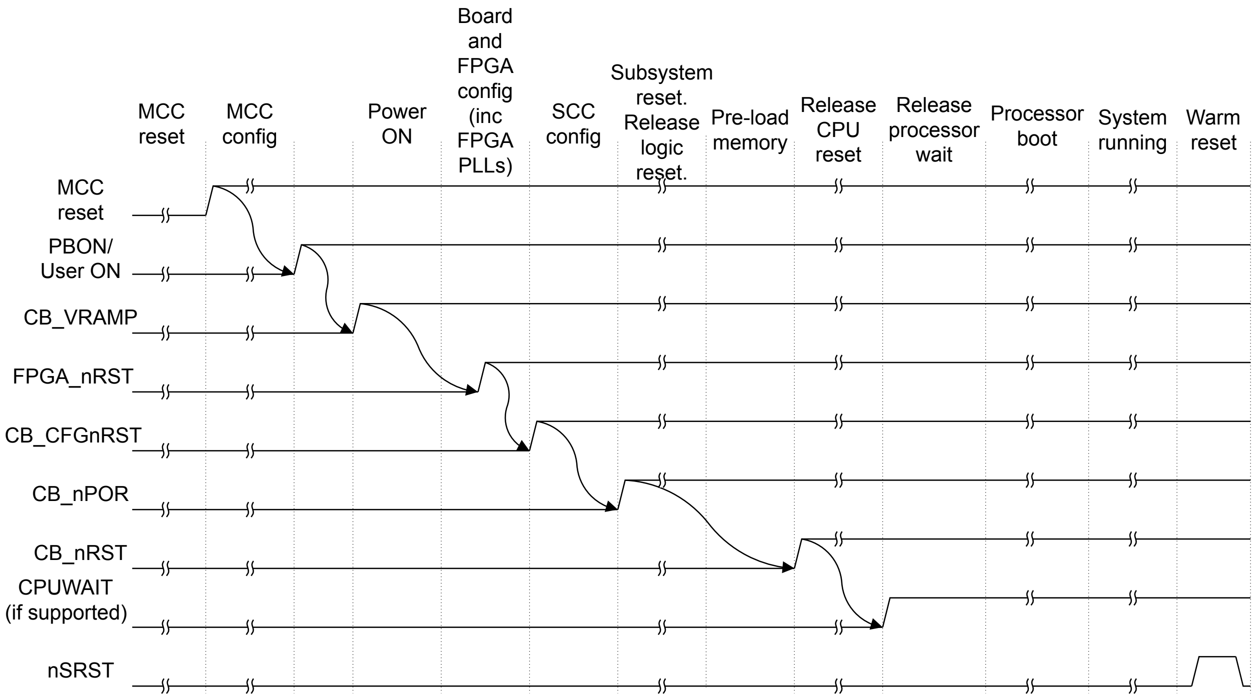


Figure 2-5 MPS3 board reset and configuration timing

Related information

1.3 Location of components on the MPS3 board on page 1-15

2.5 Power

You supply power to the board from the mains supply using the on-board power jack and the connector cable that Arm supplies with the board.

Arm supplies an external power supply unit that converts mains power to 12V 5A DC and connects to the 12V jack on the board. The unit accepts mains power in the range 110V AC to 240V AC.

——— **Caution** ———

If you supply your own external mains adapter, it must be a Low-Power Source (LPS). Some adapters are marked *LPS* on their rating label. Otherwise it might be necessary to consult you adapter supplier to ensure that it meets this criterion.

———————————

Alternatively, you can connect an external 12V DC supply, +/- 10%, directly to the 12V connector.

——— **Caution** ———

Any external 12V DC +/- 10% power supply that is used must be a limited power source, maximum 5amps.

———————————

On-board regulators supply power to the board power domains and to the FPGA power domains.

——— **Note** ———

The following on-board LEDs illuminate when power is applied:

- *12V OK*. External 12V DC connected.
- *SB_3V3 OK*. On-board 3V3 supply asserted.
- *PWR ON*. On-board 5V supply asserted.

———————————

Related information

A.9 12V power connector on page Appx-A-87

1.3 Location of components on the MPS3 board on page 1-15

2.6 Serial Configuration Controller interface

The design can include a *Serial Configuration Controller* (SCC) interface, using a block of registers in the FPGA.

After FPGA configuration, the *Motherboard Configuration Controller* (MCC) sets default values in the SCC registers, using values from the board configuration file in the configuration microSD card. The MCC configures the SCC through the serial interface on the MCC.

The following figure shows the SCC interface.

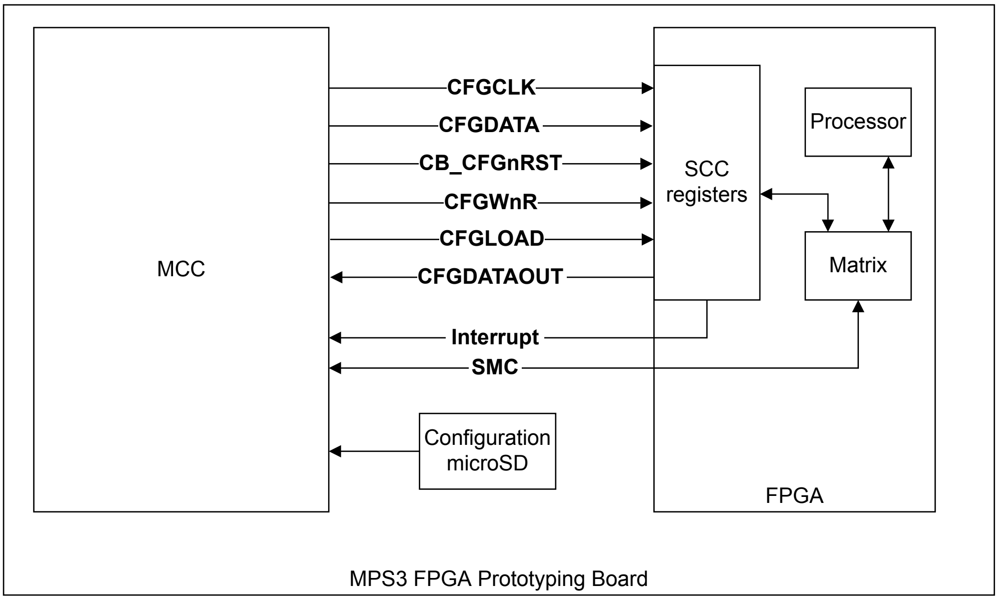


Figure 2-6 Serial Configuration Controller interface

Note

If the design does not implement the SCC, you must set the variable FPGA_SCC to FALSE in the board `config.txt` file. See *3.5.2 config.txt generic board configuration file* on page 3-64.

The following figures show the read and write cycle timing of the SCC interface.

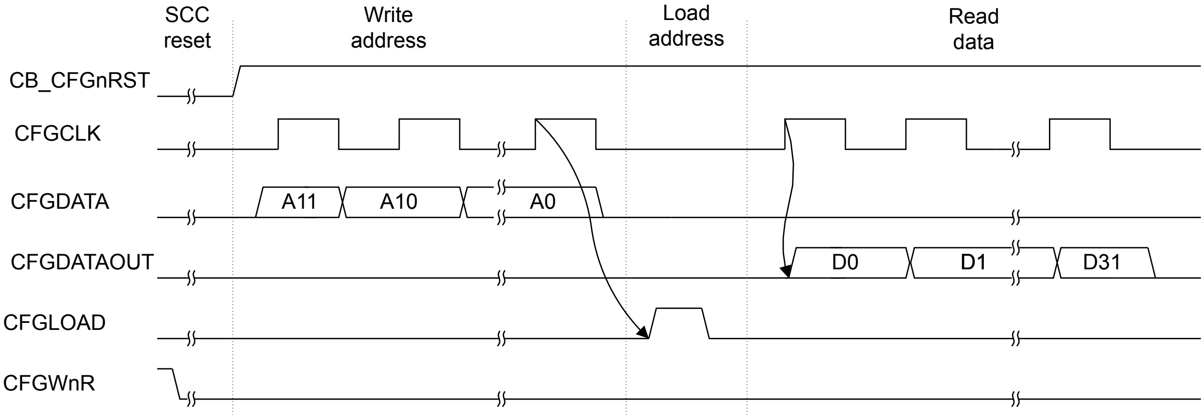


Figure 2-7 Serial Configuration Controller interface read cycle timing

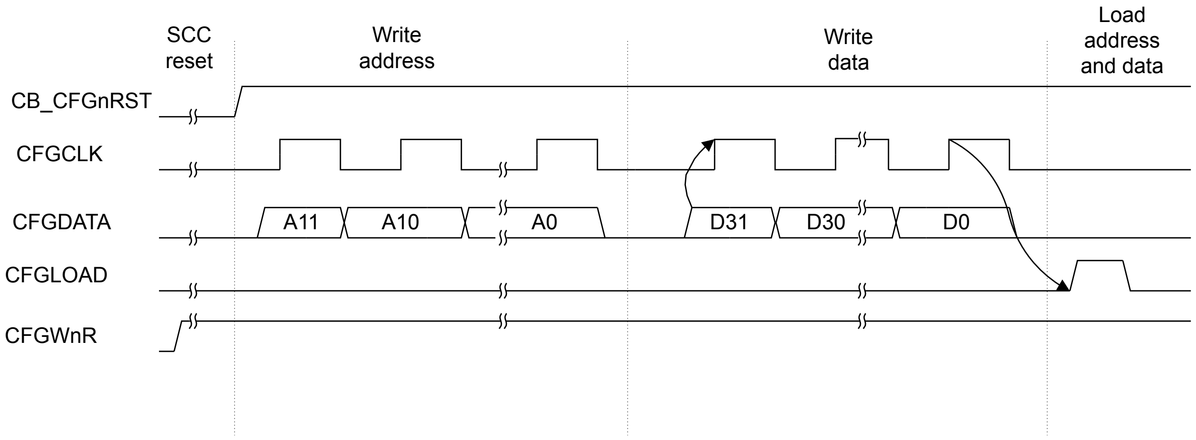


Figure 2-8 Serial Configuration Controller interface write cycle timing

2.7 MCC-SMC interface

The SMC interface in the MCC supports read and write transactions that enable communication with the internal system bus of the FPGA. The FPGA design must convert these transactions to the type of transactions that are used in the FPGA design, usually AHB-type transactions.

Overview of MCC-SMC interface

The following figure shows the FPGA-MCC SMC interface.

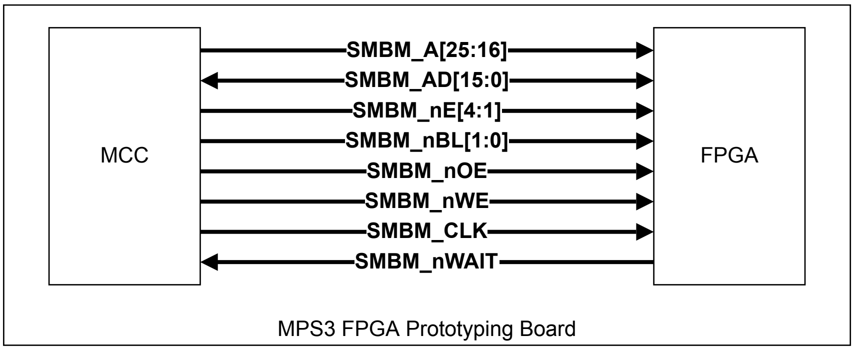


Figure 2-9 MCC-SMC interface

———— **Note** ————

SMBM_nBL[1:0] is not used.

————————————

The MCC-SMC interface enables the MCC to access the peripheral space of the FPGA system. Typical uses are:

- Preloading boot images.
- Reading and writing to system registers.
- Preconfiguring peripherals.

Implementing the MCC-SMC interface

Chip-Select

SMBM_nE[4:1] functions as a Chip-Select, providing four active-low Chip-Selects:

- `0xF`: No Chip Select.
- `0xE`: Chip-Select 0.
- `0xD`: Chip-Select 1.
- `0xB`: Chip-Select 2.
- `0x7`: Chip-Select 3.

Address and data transfer out of the MCC-SMC interface

The MCC provides 25 of the 32 bits that the FPGA design uses for AHB-type transactions. The FPGA design generates the other 7 bits.

During the address phase, the 25 address bits are shared between **SMBM_D[15:0]** and **SMBM_A[24:16]**.

- **SMBM_D[15:0]** carries the 16 least significant address bits of the interface.
- **SMBM_A[24:16]** carries the nine most significant address bits of the interface.

———— **Note** ————

SMBM_A[25] is not used.

————————————

During the data transfer phase, **SMBM_D[15:0]** carries the four data bytes in two stages. It carries the two least significant bytes, and then the two most significant data bytes.

The following figure shows an MCC-SMC interface address and data transfer.

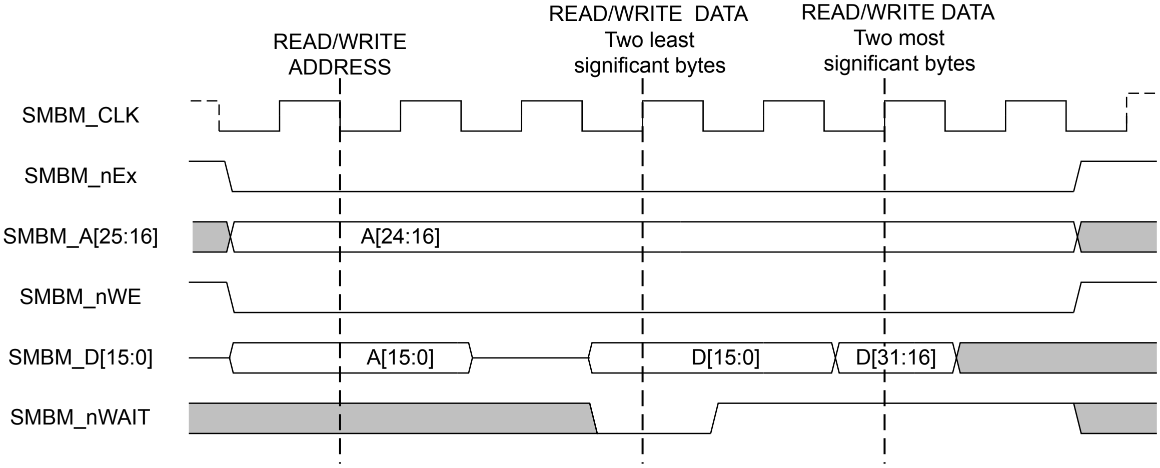


Figure 2-10 MCC-SMC interface address and data transfer timing

———— **Note** ————

- The MCC-SMC interface supports only 32-bit data read and write transfers.
- To meet the timing requirements for address and data transfer, Arm recommends that you use the MCC-SMC interface decoder in the file `microToAhb.v`, provided by Arm.

————————————

Forming the 32-bit address in the FPGA

The FPGA design must form the 32-bit address, for AHB- type transfers inside the FPGA, from the following:

- The least significant bit (LSB), generated by the FPGA design.

  Note

  The MCC-SMC interface supports only four-byte address mode transactions. To support four-byte address mode, the LSB generated by the FPGA design must be `0b0` and the LSB of the MCC-SMC interface is always `0b0`.

- 25 address bits from the MCC.
- Six user bits, generated by the FPGA design.

The MCC can access 64MB of user memory for each Chip-Select, that is, a total memory space of 256MB. Each Chip-Select can point to non-contiguous areas in the user design. But the total amount of user memory that each Chip-Select accesses cannot exceed 64MB.

The six address bits generated by the design, and if necessary, the Chip-Select bits, define which parts of the user memory space are accessed.

The following figure shows the formation of the AHB 32-bit address in the FPGA.

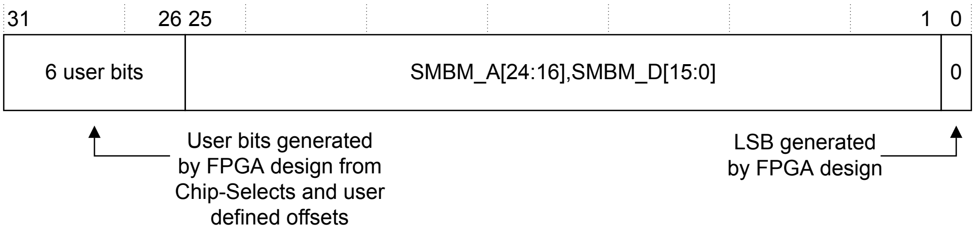

31
26
25
1
0
6 user bits
SMBM_A[24:16],SMBM_D[15:0]
0
User bits generated by FPGA design from Chip-Selects and user defined offsets
LSB generated by FPGA design

Figure 2-11 Formation of 32-bit address in FPGA

Example user memory map

The following figure shows an example user memory map.

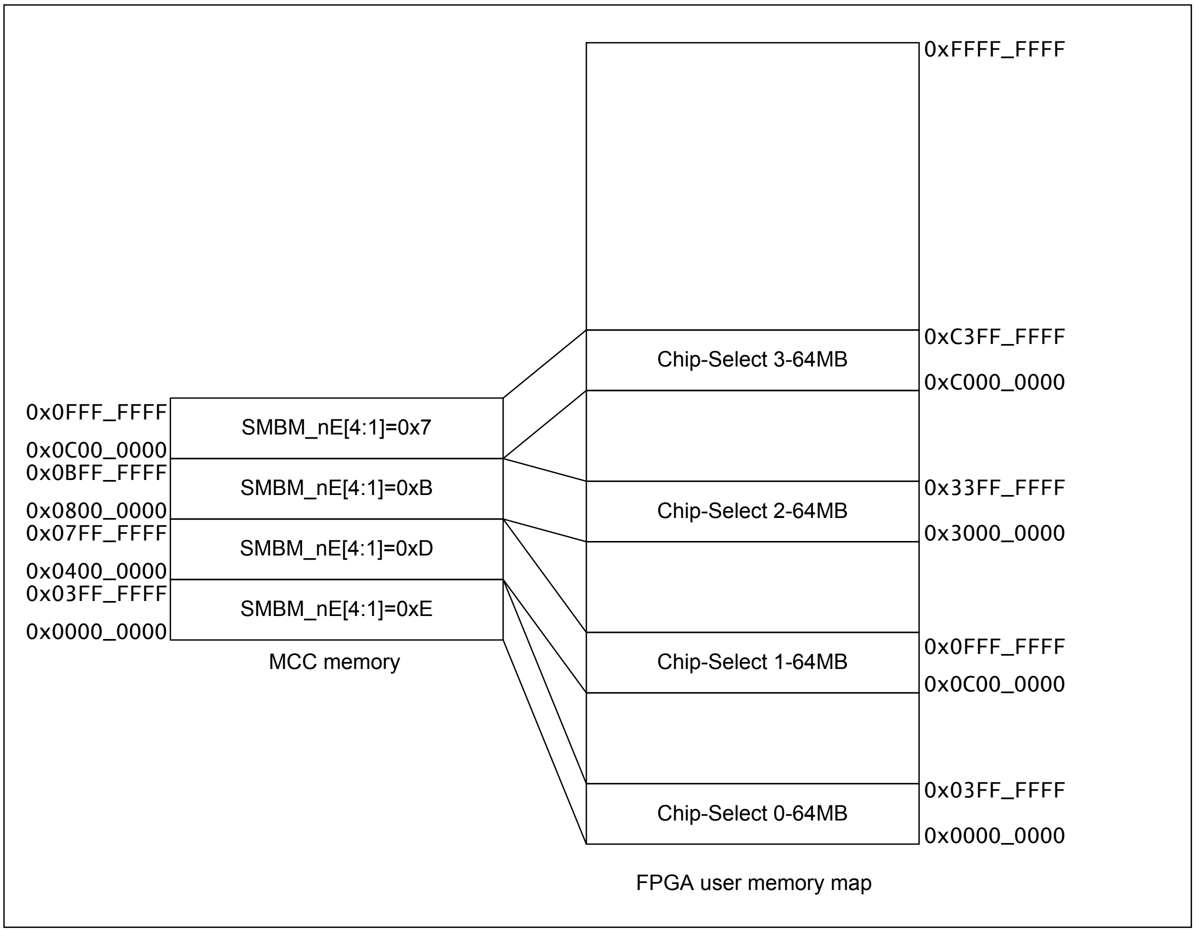


Figure 2-12 Example user memory map

The example memory map shows the mapping when the six user bits have the following values for each Chip-Select:

- CS0:
  — The image in the FPGA generates internal AHB address bits[31:26] =`0b000000` which gives a base address of `0x00000000`.
- CS1:
  — The image in the FPGA generates internal AHB address bits[31:26] =`0b000011` which gives a base address of `0x0C000000`.
- CS2:
  — The image in the FPGA generates internal AHB address bits[31:26] =`0b001100` which gives a base address of `0x30000000`.
- CS3:
  — The image in the FPGA generates internal AHB address bits[31:26] =`0b110000` which gives a base address of `0xC0000000`.

The variable IMAGE0ADDRESS in the `images.txt` configuration file defines the base address in the MCC memory. In this example, setting IMAGE0ADDRESS to `0x05000000` selects a base address of `0x0D000000` in the FPGA user memory. The firmware sets **SMBM_nE[4:1]** to `0xD` to select Chip-Select 1 which maps to the correct area of user memory.

See *3.5.3 Contents of the MB directory* on page 3-64 for information on the `images.txt` file.

MCC-SMC interface not implemented

If the design does not implement the MCC-SMC interface, you must set the variable FPGA_SMB to FALSE in the board `config.txt` file. See *3.5.2 config.txt generic board configuration file* on page 3-64.

2.8 USB 2.0 and Ethernet static memory interface

The FPGA incorporates a simple *Static Memory Controller* (SMC) that interfaces to the USB 2.0 and Ethernet controllers on the MPS3 board.

The following figure shows the USB 2.0 and Ethernet SMC interface.

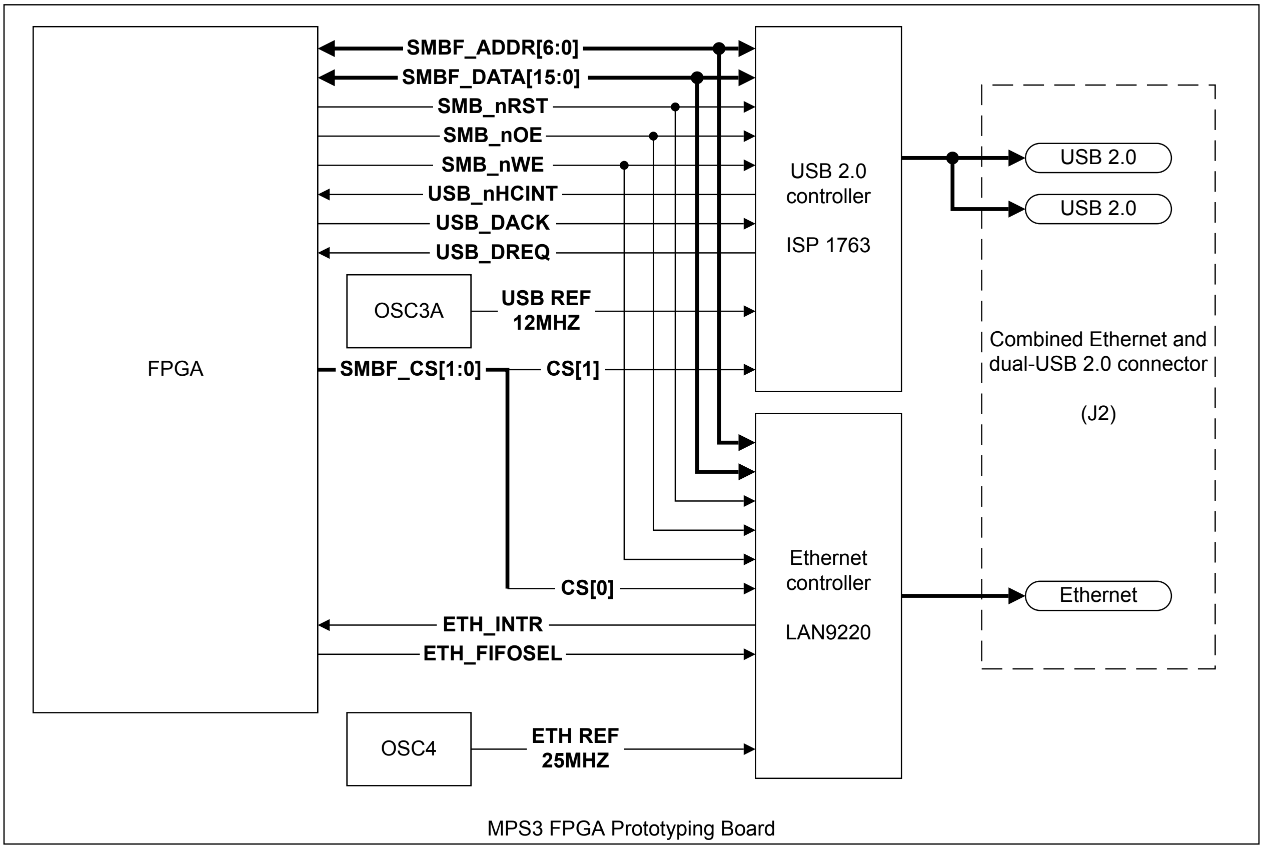


Figure 2-13 MPS3 board USB 2.0 and Ethernet static memory interface

Related information

A.6 Combined Ethernet and dual USB-A connector on page Appx-A-84

1.3 Location of components on the MPS3 board on page 1-15

2.9 Video HDLCD interface

The HDMI controller and HDMI connector on the MPS3 board enable you to implement an HDLCD interface.

The external controller is a frame buffer device that can display up to 1920 × 1080p resolution at 60fps.

Note

Support for higher resolutions up to 1080p depends on the timing performance of your FPGA image.

The HDMI controller also supports I^{2}S audio from the FPGA.

The following figure shows a video HDLCD interface example design.

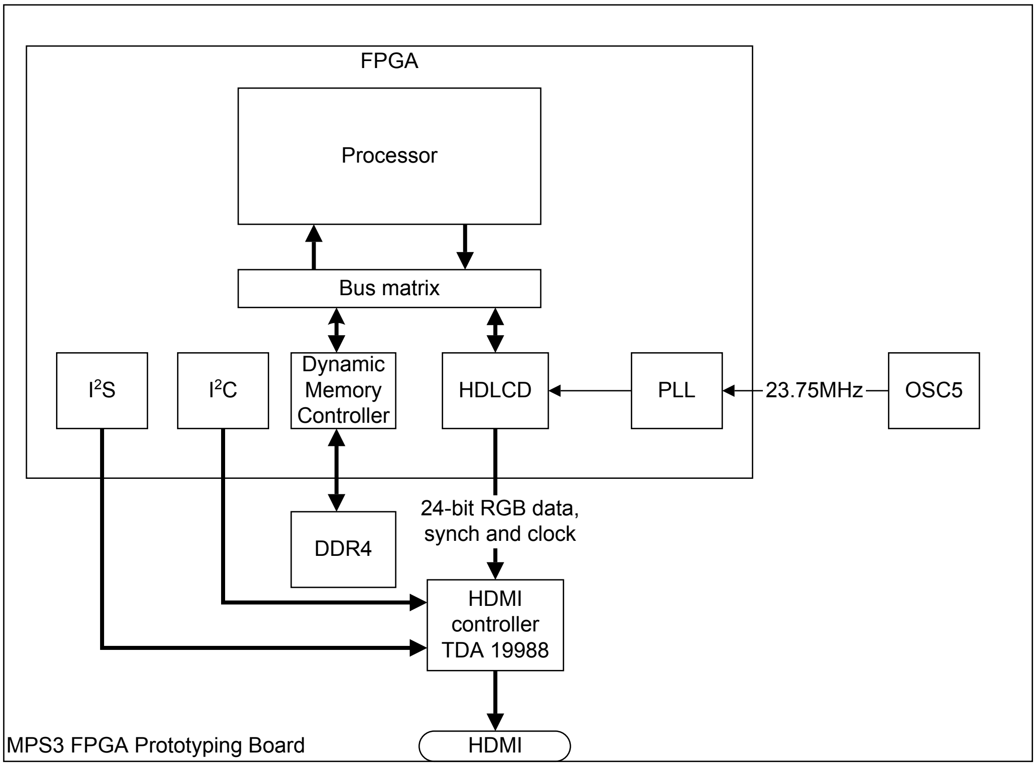

FPGA
Processor
Bus matrix
I²S
I²C
Dynamic Memory Controller
HDLCD
PLL
23.75MHz
OSC5
DDR4
24-bit RGB data, synch and clock
HDMI controller TDA 19988
MPS3 FPGA Prototyping Board
HDMI

Figure 2-14 MPS3 board video HDLCD interface example design

Related information

A.7 HDMI Type A female connector on page Appx-A-85

1.3 Location of components on the MPS3 board on page 1-15

2.10 Audio codec interface

An AACI audio codec on the MPS3 board provides a stereo audio interface with Line In, Line Out, and Microphone In.

The AACI audio codec and the stereo audio interface enable you to implement an audio codec interface.

The FPGA configures the codec over I^2C and has an I^2S audio interface. The I^2S controller is a dedicated controller that consists of a data buffer and serializer.

The interface supports the standard audio data rate of 48kHz, up to a maximum of 96kHz. The codec contains audio power amplifiers that can drive up to 500mW, 8Ω stereo speakers.

The audio codec drives the stacked stereo jack on the MPS3 board. When using an electret type of microphones, use jumpers J58 (L) and J59 (R) to enable microphone bias current.

The following figure shows an audio codec interface example design.

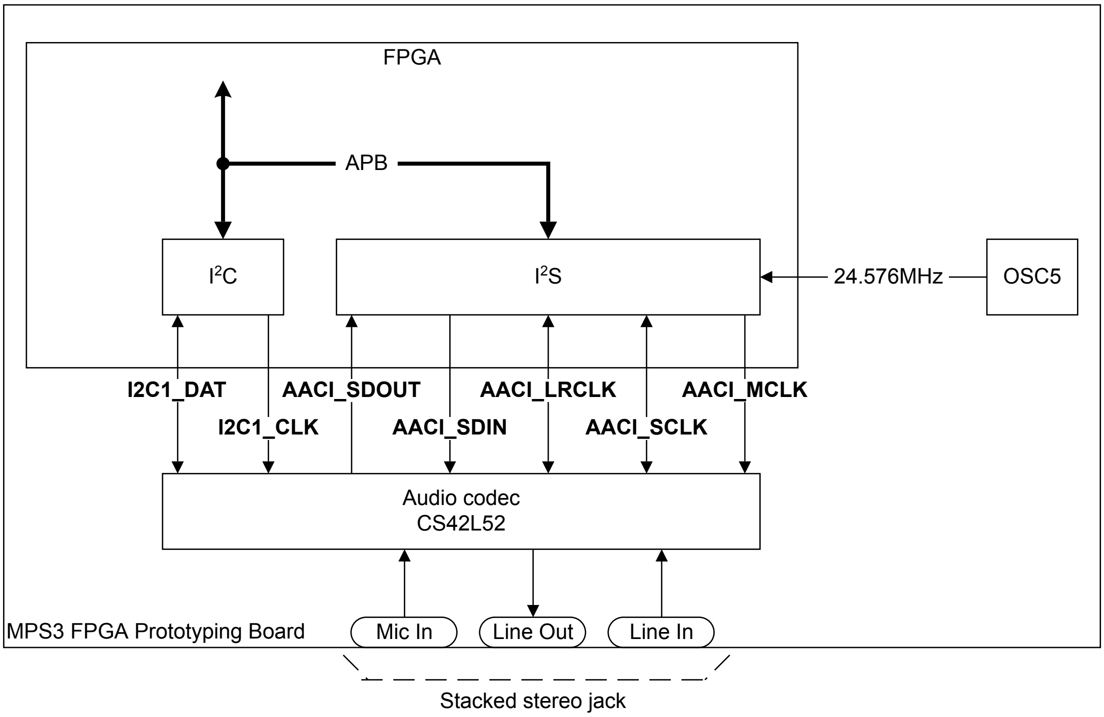


Figure 2-15 MPS3 board audio codec interface example design

Related information

A.8 Audio connectors, stacked stereo jacks on page Appx-A-86

1.3 Location of components on the MPS3 board on page 1-15

2.11 QVGA video CLCD display

The MPS3 board provides QVGA, 320 × 240, CLCD video display.

The CLCD video display system provides an on-board CLCD display panel that includes:

- An 8-bit parallel bus between the FPGA and the display panel.
- A 4-wire resistive touch screen.
- A *Touch Screen Controller* (TSC) that connects to the FPGA over an I^{2}C bus.

The interrupt signal from the touch screen controller, **LCD_TSINT**, and the backlight control signal, **LCD_BLC**, connect to the GPIO interface in the FPGA.

The interface supports a screen update rate of 20fps.

The following figure shows a functional overview of the CLCD display system.

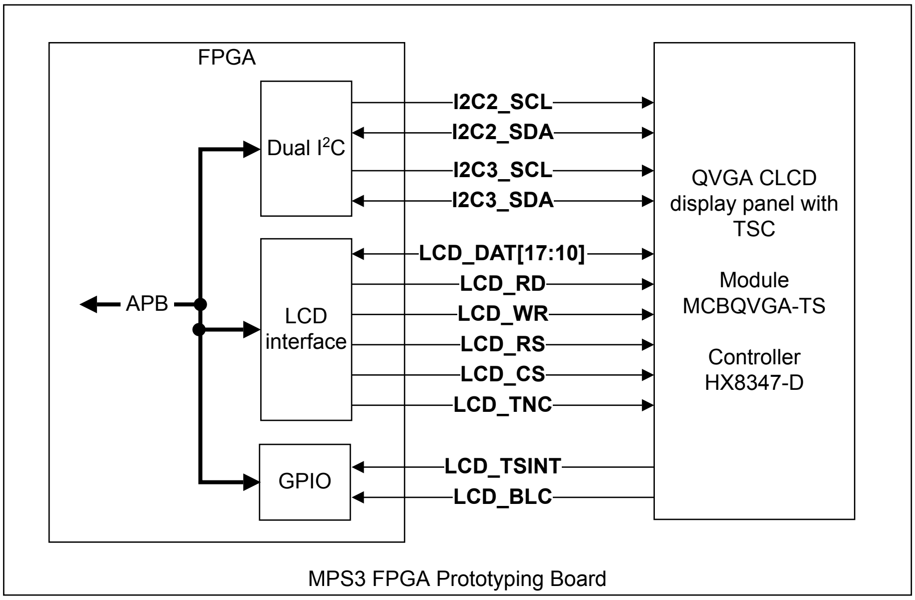


Figure 2-16 MPS3 board CLCD display system

2.12 On-board user components

The MPS3 board provides ten user LEDs, eight user switches, and two user push buttons.

The LEDs, switches, and push buttons connect directly to the FPGA, meaning they can be used for debug.

The following figure shows the user components on the MPS3 board.

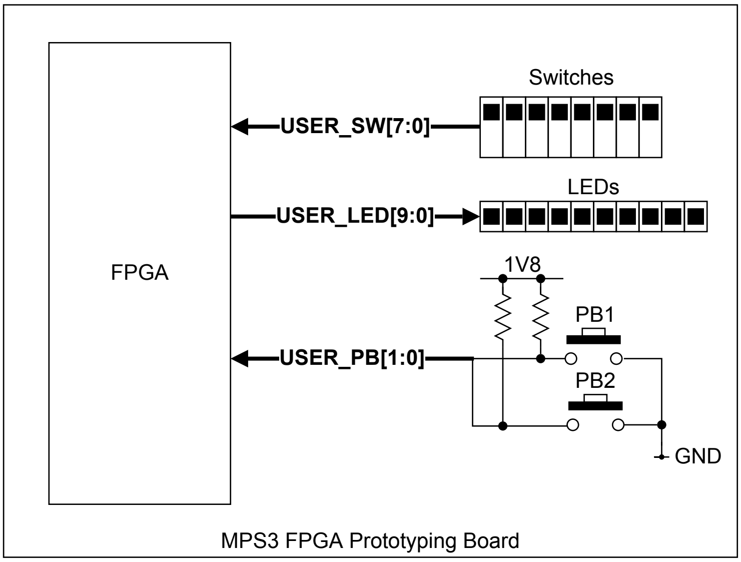


Figure 2-17 MPS3 board user components

Related information

1.3 Location of components on the MPS3 board on page 1-15

2.13 Interrupts

Interrupt signals from peripherals on the MPS3 board connect to external pins on the FPGA.

The image that you implement connects the peripheral interrupts to systems in the FPGA.

The following table shows the peripheral interrupt signals that connect to external pins on the FPGA.

Table 2-2 Peripheral interrupts

Interrupt	Interrupt source
IOFPGA_SYSWDT	RTCC/SYS_CFG (OUT)
PB_IRQ	Push button
WDOG_RREQ	Watchdog reset (OUT)
DVI_INT	HDMI interrupt
USB_INT	USB 2.0
ETH_INT	Ethernet
CLCD_TINT	CLCD touchscreen

2.14 FPGA DDR4 memory interface

The MPS3 board provides 4GB of DDR4 SODIMM and a DDR4 interface to the FPGA.

The DDR4 controller and PHY interface uses the Xilinx *Memory Interface Generator* (MIG). The interface is 64-bit and uses 10 byte lanes.

The interface supports up to 900MHz, 1800MT/s, with the supplied SODIMM (MTA4ATF51264HZ-2GB31).

The following figure shows the FPGA DDR4 memory interface.

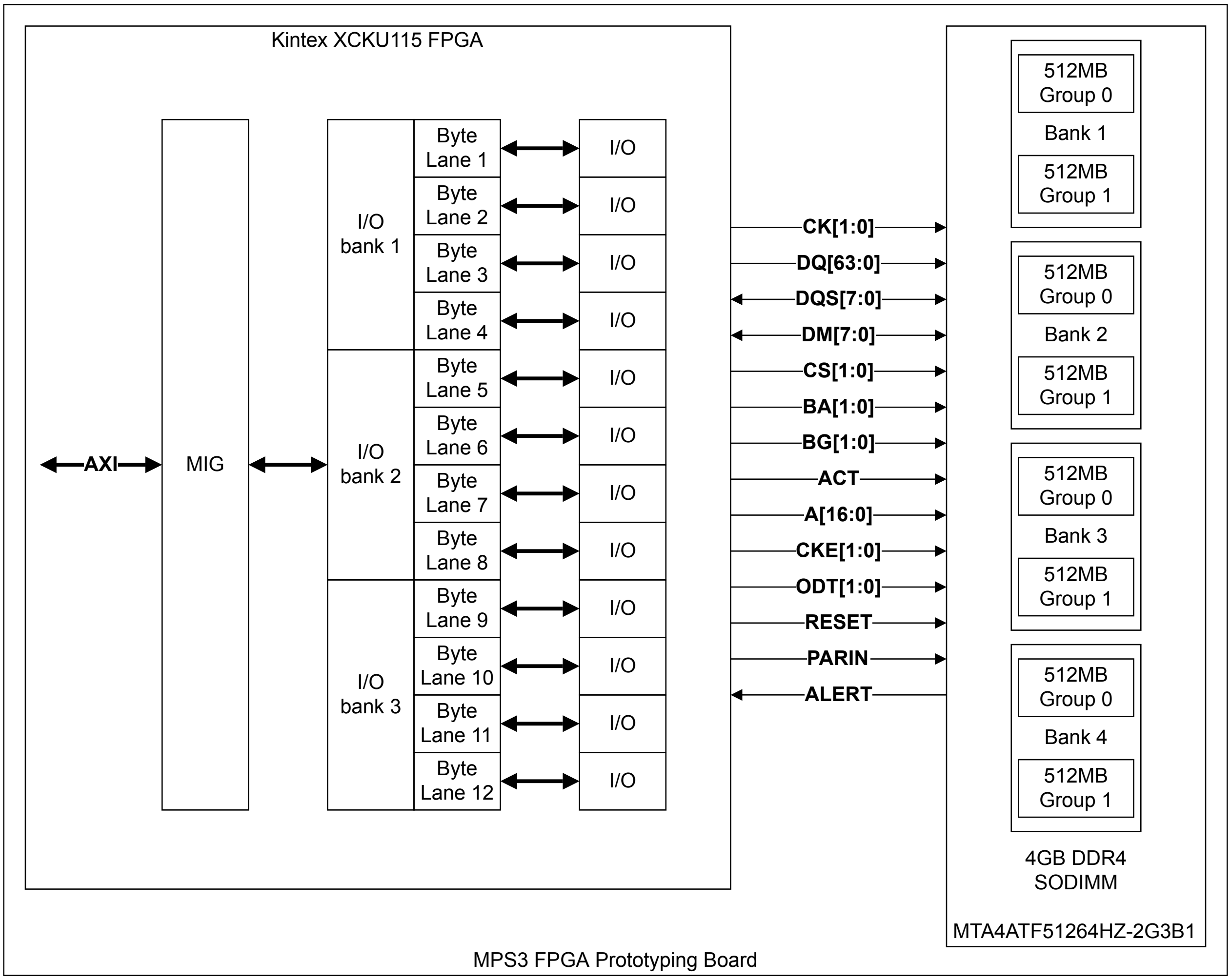


Figure 2-18 Kintex XCKU115 FPGA DDR4 memory interface

2.15 User non-volatile memory

The MPS3 board provides on-board user non-volatile memory, 8MB QSPI flash (SST26VF064B), 16GB eMMC16G_M525, and a microSD card interface.

A typical use of the QSPI flash is as boot memory. The microSD card or eMMC memory can be used for storing the Linux file system.

The following figure shows a non-volatile memory system example design. For completeness, the figure includes the DDR4 volatile memory.

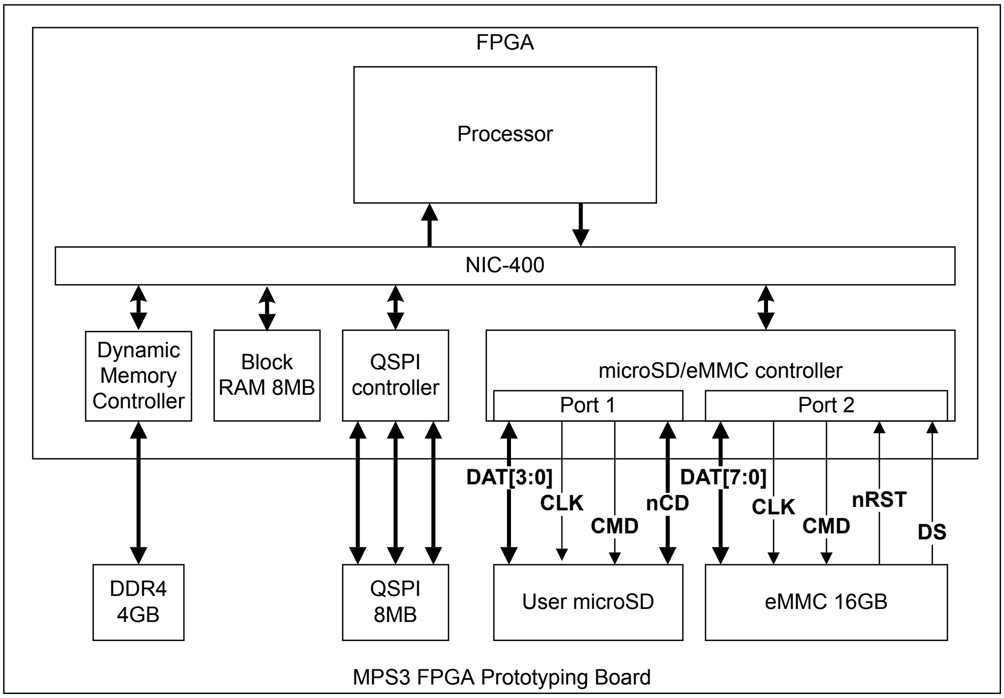


Figure 2-19 MPS3 board non-volatile memory system example design

Note

The simplest boot method is to use block RAM in the FPGA which can be pre-loaded by the MCC before resets are released. The use of block RAM requires the FPGA design to implement theMCC-SMC interface. See *2.7 MCC-SMC interface* on page 2-30.

2.16 Arduino Shield and Pmod interfaces

The MPS3 board supports peripheral development by providing two Arduino Shield interfaces and, as an alternative, four Peripheral Module (Pmod) interfaces (Type 2A/3/4 support).

Overview of Shield and Pmod interfaces

The following figure shows the two Shield interfaces, and the four Pmod interfaces where the FPGA design implements an SPI controller that drives the 12-bit ADC.

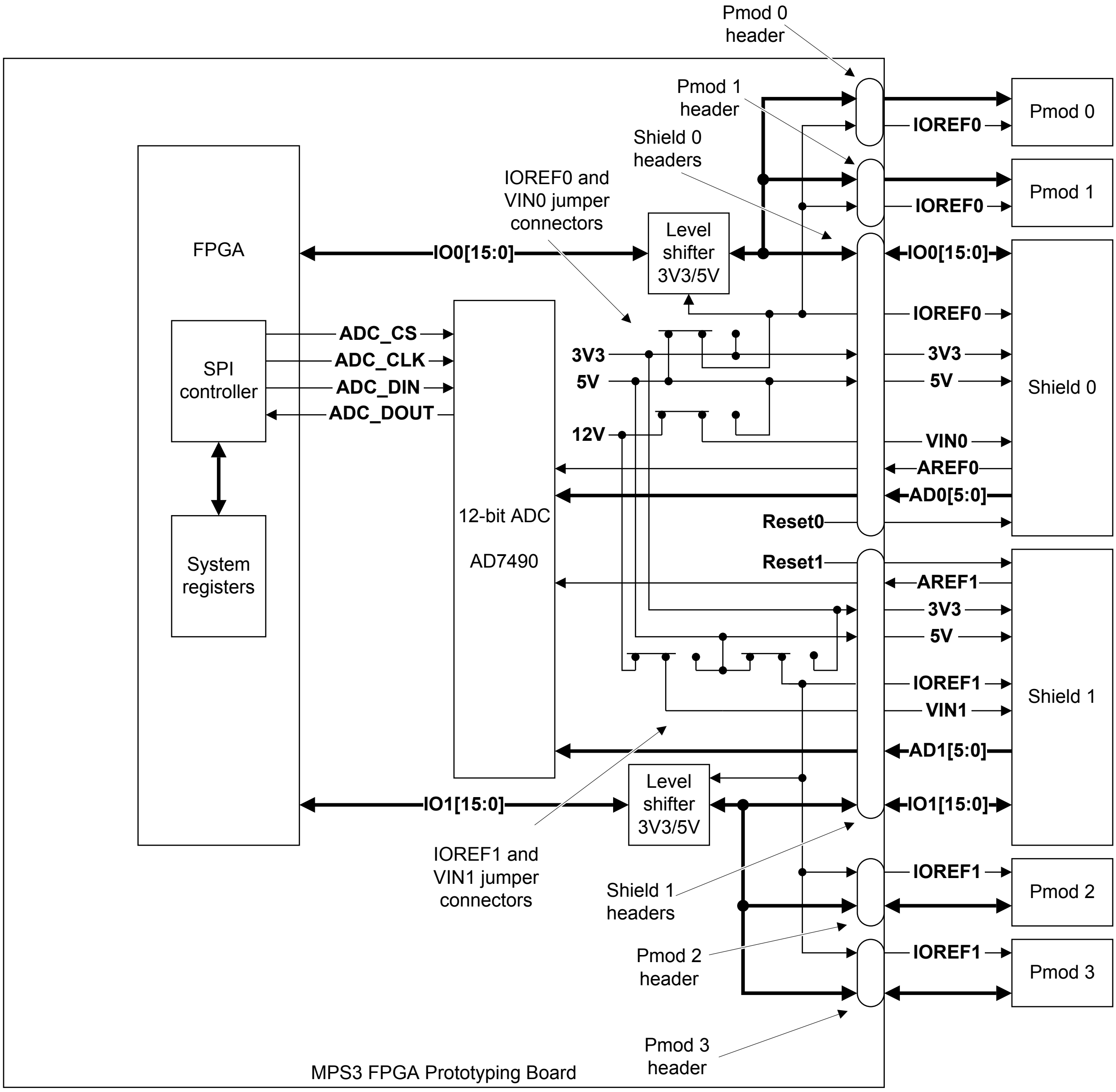


Figure 2-20 Shield and Pmod interfaces

The Arduino Shield and Pmod interfaces share pins on the FPGA.

The Shield and Pmod interfaces share signals, including the digital I/O. Each Shield interface has 16 digital I/O, and the four Pmod interfaces each have 8 digital I/O. See the following sections for information on how the Shield and Pmod interfaces share the digital I/O signals:

- *A.2 Arduino Shield connectors* on page Appx-A-76.
- *A.3 Peripheral Module (Pmod) connectors* on page Appx-A-80.

The interfaces enable the use of off-the-shelf sensor interfaces, for example, WiFi, Bluetooth, proximity sensors, and gyro sensors, to suit custom design requirements. The interfaces also enable you to add full-custom sensor modules to the system.

Shield interfaces

Each Shield interface has 16 I/O and supports the choice of 3V3 or 5V I/O, independently selectable for each Shield using two user-links.

The outputs from the digital I/O user-links also act as the digital I/O voltage references for the Shields.

Other links, one for each Shield, select 5V or 12V for the power inputs, VIN0 and VIN1.

Note

The links also select I/O references and power inputs to the Pmod interfaces.

See *Shield and Pmod power and I/O reference voltage user-links* on page 2-44 for more information on the use of the user-links.

A 12-bit ADC supports six analog channels on each Shield.

Pmod interfaces

The Pmod interfaces are an alternative to the Shield interfaces. Each Pmod expansion header has 8 digital I/O and supports the choice of 3V3 or 5V digital I/O. You can select the digital I/O using two user-links on the board. One user-link selects 3V3 or 5V digital I/O operation for Pmod 0 and Pmod 1. The other user-link selects 3V3 or 5V operation for Pmod 2 and Pmod 3.

The outputs from the digital I/O user-links also act as the digital I/O voltage references for the Pmod expansion boards.

Two other links select 5V or 12V for the power inputs, VIN0 and VIN1.

Note

The links also select digital I/O references and power inputs to the Shield interfaces.

See *Shield and Pmod power and I/O reference voltage user-links* on page 2-44 for more information on the use of the user-links.

The locations of the Pmod connectors support the use of a dual-connector board, if required.

The Pmod interfaces do not support analog I/O.

Shield and Pmod power and I/O reference voltage user-links

The following figure shows the user-links that select I/O voltage level and power inputs to the Shield and Pmod interfaces. The figure shows the user-links that select 12V or 5V power, and 5V or 3V3 I/O references, for the Shields.

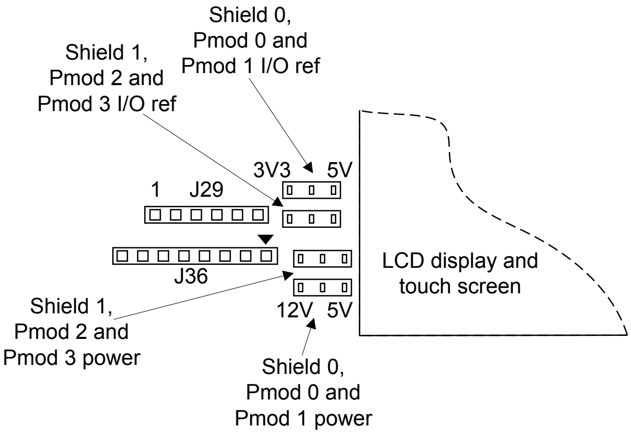


Figure 2-21 Shield and Pmod power and I/O reference voltage user-links

One user-link selects 3V3 or 5V I/O operations for the Shield 0, Pmod 0, and Pmod 1 interfaces.

One user-link selects 3V3 or 5V I/O operations for the Shield 1, Pmod 2, and Pmod 3 interfaces.

One user-link selects 5V or 12V power for the Shield 0 interface.

One user-link selects 5V or 12V power for the Shield 1 interface.

———— **Caution** ————

The maximum currents available at the power and reference pins are:

3V3/IOREF	1A maximum available for both Shields and all four Pmod interfaces.
5V/IOREF	1A maximum available for both Shields and all four Pmod interfaces.
12V	0.5A maximum available for both Shields.

————————————

See *1.3 Location of components on the MPS3 board* on page 1-15 for the location of the user-links.

Related information

A.2 Arduino Shield connectors on page Appx-A-76

A.3 Peripheral Module (Pmod) connectors on page Appx-A-80

1.3 Location of components on the MPS3 board on page 1-15

2.17 FMC-HPC interface

The MPS3 board supports high speed, high pin count expansion using the *FPGA Mezzanine Card* (FMC) standard.

Overview of the FMC-HPC interface

The MPS3 board uses the *FMC-High Pin Count* (FMC-HPC) variant and provides:

- One 400-way Samtec SEARAY connector.
- 160 I/O.
- 10 × high-speed differential transceivers, *Multi-Gigabit Transceivers* (MGTs).
- 2 × high-speed differential clocks, MGT clocks.
- Four differential clocks.
- Power.

The following figure shows the MPS3 board FMC-HPC interface.

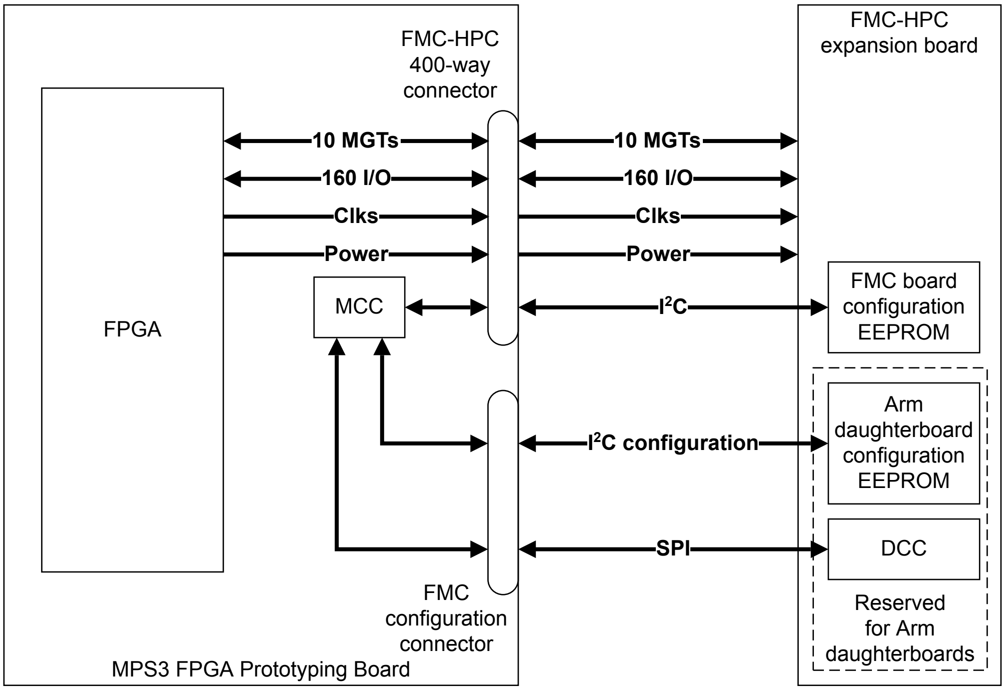

FMC-HPC
400-way
connector
FMC-HPC
expansion board
10 MGTs
160 I/O
Clks
Power
10 MGTs
160 I/O
Clks
Power
FPGA
MCC
I²C
FMC board configuration EEPROM
I²C configuration
Arm daughterboard configuration EEPROM
SPI
DCC
Reserved for Arm daughterboards
FMC configuration connector
MPS3 FPGA Prototyping Board

Figure 2-22 MPS3 board FMC-HPC interface

Note

The FMC board configuration EEPROM is part of the FMC standard. The Arm daughterboard configuration EEPROM and *Daughterboard Configuration Controller* (DCC) are not part of the FMC standard but are supplied with Arm FMC daughterboards.

FPGA-FMC pin connectivity

The FPGA-FMC pin connectivity is available in *Application Note AN533 Blinky example FPGA image for the MPS3 Prototyping Board*.

The Application Note can be downloaded as a pack from:

https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/download-fpga-images.

The pack contains the following key folders:

- `pcb`:
  - — Contains the schematics and PCB top/bottom gerber plots for the board.
- `logical`:
  - — Contains the RTL source and the constraints file.
- `docs`. Contains the following files:
  - — *V2M_MPS3_fmc_pinout.xlsx*. This spreadsheet shows the MPS3 FMC pinout and net names.
  - — *V2M_MPS3_fpga_pinout.xlsx*. This spreadsheet shows the MPS3 FPGA pinout and net names.
  - — `revision history.txt`
  - — *Application Note AN533 Blinky example FPGA image for the MPS3 Prototyping Board* PDF document.
- `boardfiles`:
  - — Contains the FPGA image (`.bit`) file and the MCC firmware (`.ebf`) file that must be loaded onto the MPS3 microSD card to run the AN533 blinky design on the MPS3 board.

Example FMC board layout

The following figure shows an MPS3 board with an FMC board that is fitted to the FMC-HPC expansion connector.

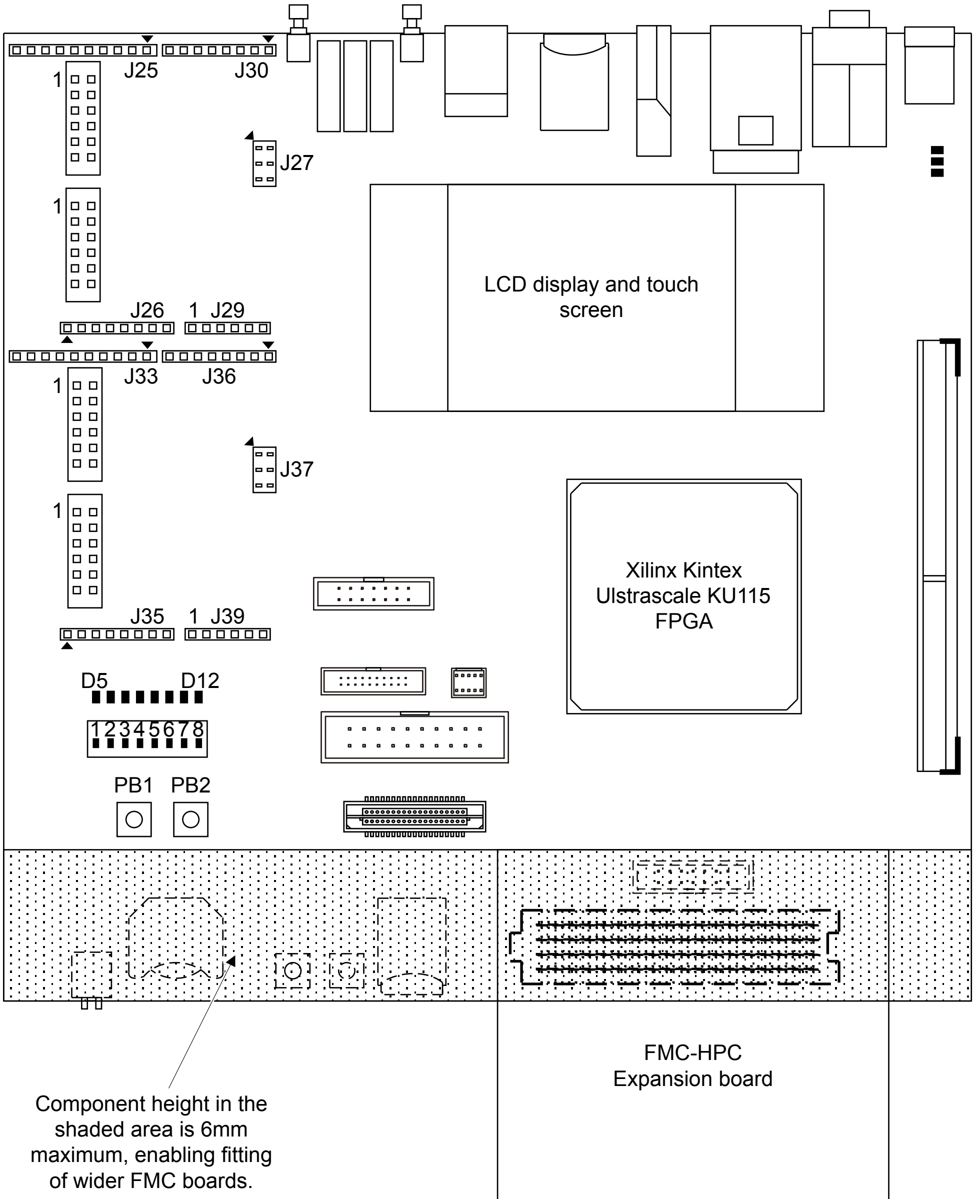

J25
J30
1
J27
1
J26
1 J29
J33
J36
1
J37
1
J35
1 J39
D5
D12
1 2 3 4 5 6 7 8
PB1
PB2
LCD display and touch screen
Xilinx Kintex Ulstrascale KU115 FPGA
FMC-HPC Expansion board
Component height in the shaded area is 6mm maximum, enabling fitting of wider FMC boards.

Figure 2-23 FMC-HPC board fitted to the MPS3 board FMC-HPC expansion connector

The figure shows a single-width FMC board, 69mm × 76.5mm, fitted. The maximum component height in the shaded area in the figures is 6mm to enable fitting of wider FMC boards.

Configuration of Arm FMC expansion boards

The MPS3 board provides a custom 14-pin connector to enable configuration of Arm FMC boards using I^2C, SPI, and reset signals. The configuration process is similar to the configuration process used on the Arm Versatile Express boards that use the signals on the HDRY headers.

Arm FMC expansion boards provide a daughterboard EEPROM and a *Daughterboard Configuration Controller* (DCC). The DCC and the MCC on the MPS3 board configure the Arm FMC expansion board using the configuration information in the daughterboard EEPROM.

Examples of available third-party FMC expansion boards

The following are examples of FMC third-party expansion boards that are available to support various interfaces:

ADC board
AD9467-FMC.

Loopback board
WHZ-FMC XM-107.

Related information

A.4 FMC-HPC connector on page Appx-A-82

A.5 FMC configuration connector on page Appx-A-83

1.3 Location of components on the MPS3 board on page 1-15

2.18 System debug

The MPS3 board provides several methods of performing debug.

CoreSight™ debug

The following figure shows the MPS3 board debug and trace system.

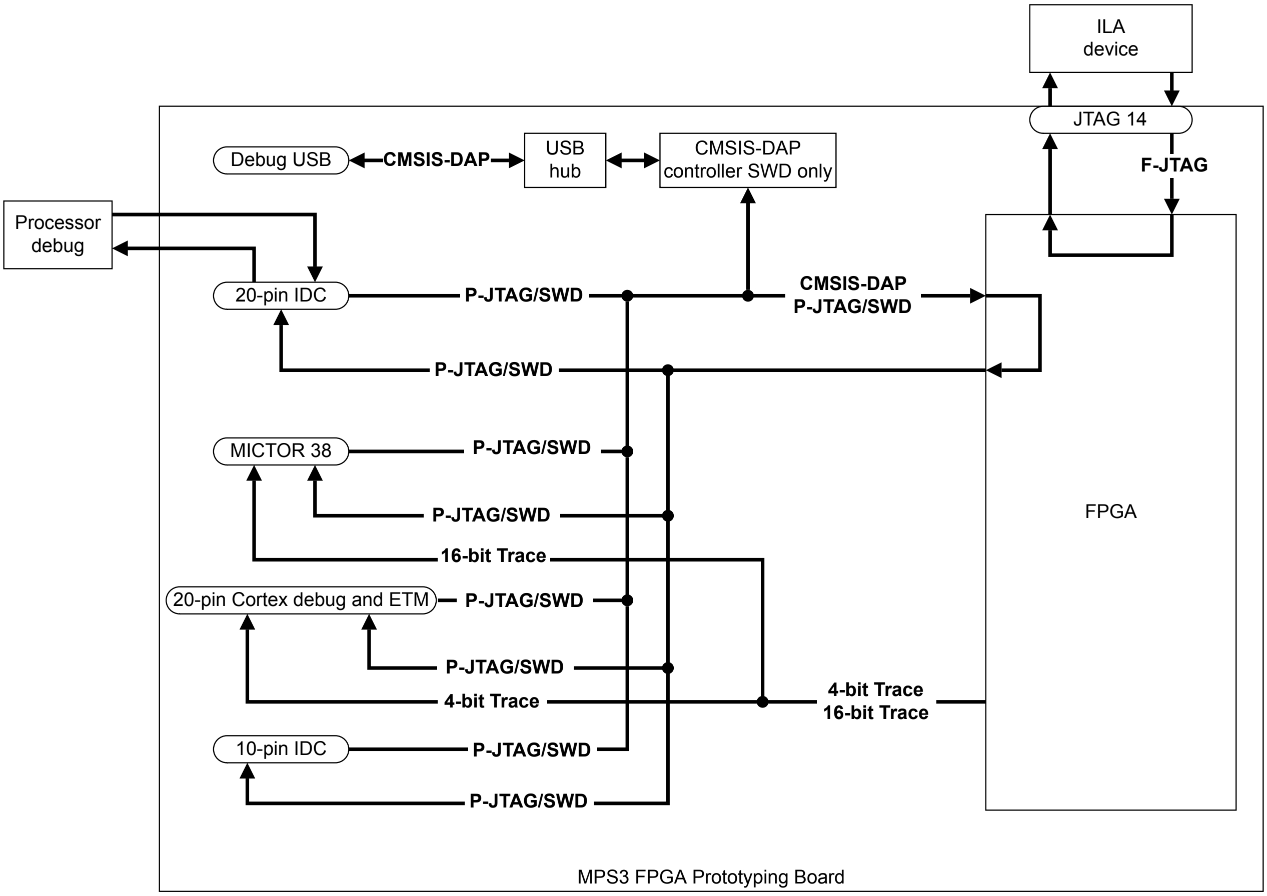


Figure 2-24 MPS3 board CoreSight debug and trace

- P-JTAG processor debug on:
  - — 20-pin IDC connector.
  - — 10-pin IDC connector.
  - — 20-pin Cortex debug and ETM connector.
  - — 38-pin MICTOR connector.
- *Serial Wire Debug* (SWD) on:
  - — 20-pin IDC connector.
  - — 10-pin IDC connector.
  - — 20-pin Cortex debug and ETM connector.
  - — 38-pin MICTOR connector.
  - — CMSIS-DAP debug over USB on the Debug USB connector, USB 2.0 type B connector.
- 16-bit trace on a 38-pin MICTOR connector.
- 4-bit trace on a 20-pin Cortex debug and ETM connector.
- FPGA debug on 14-pin ILA connector for FPGA debug.

——— **Note** ———

- The availability of P-JTAG, SWD, 16-bit trace, and 4-bit trace, depends on the design that you implement in the FPGA.
- The Debug USB connector supports remote USB access to the MCC to enable remote reboot, remote reset, and remote shutdown.

Debug over USB

Debug over USB supports:

- Four UARTs.
- MCC debug.
- CMSIS-DAP.

The following figure shows the architecture of the debug over USB system.

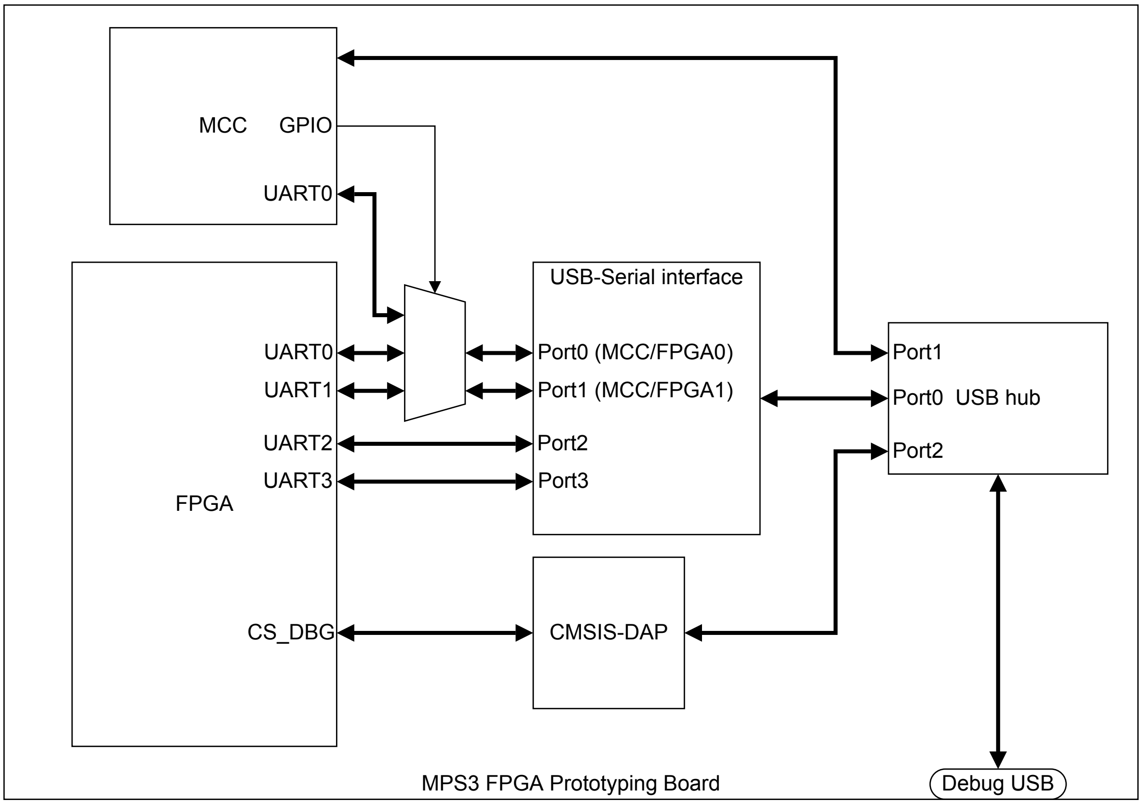


Figure 2-25 MPS3 board debug over USB system architecture

Serial ports

The four user serial ports on the Kintex XCKU115 FPGA indicate the status of the system during boot time. After the system has booted, they are used for debug or status information. The four serial ports are concentrated into a USB-serial interface that connects to a four port hub, that is connected to the USB connector. The hub enables the four FPGA serial ports, the MCC USBDBG port, and the CMSIS-DAP controller to share the USB debug connector.

CMSIS-DAP debug

The CMSIS-DAP interface in the FPGA enables debug over USB to the FPGA CoreSight components using the dedicated Arm CMSIS-DAP controller.

Setting up host software for the hub and serial ports

The MPS3 board does not require software drivers for Windows 7 or later versions of Windows.

The following figure shows the USB devices that the MPS3 board adds when you connect a workstation to the USB debug connector.

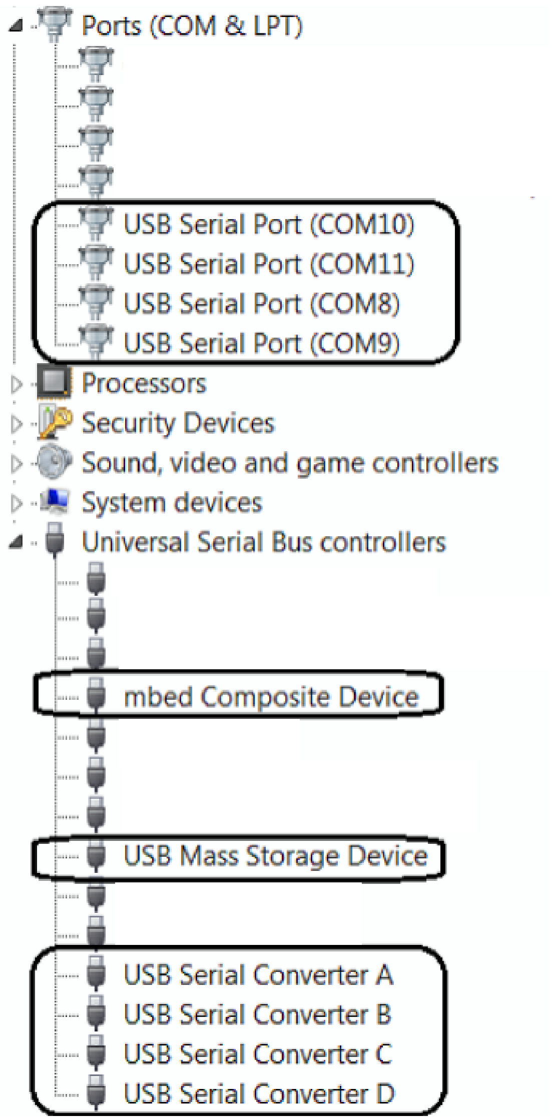


Figure 2-26 MPS3 board software drivers

Related information

A.1.1 20-pin IDC connector on page Appx-A-71

A.1.2 10-pin IDC connector on page Appx-A-72

A.1.3 20-pin Cortex debug and ETM connector on page Appx-A-72

A.1.4 38-pin MICTOR connector on page Appx-A-73

A.1.5 14-pin F-JTAG ILA connector on page Appx-A-74

A.1.6 Debug USB 2.0 connector on page Appx-A-75

1.3 Location of components on the MPS3 board on page 1-15

2.19 Design settings for correct board operation with a minimal design

For correct operation with a minimal design, the MPS3 board requires a minimum amount of RTL in the FPGA, and certain variable settings in the `config.txt` file.

Minimum RTL

The following table shows the signals that you must tie off in the FPGA to generate the minimum RTL for correct operation of the MPS3 board.

Table 2-3 Minimum RTL for correct operation of the MPS3 board

FPGA signal	Minimum RTL
MMB_IDCLK	Tie LOW
EMMC_CLK	Tie LOW
QSPI_nCS	Tie HIGH
QSPI_SCLK	Tie LOW
IOFPGA_SYSWDT	Tie LOW
WDOG_RREQ	Tie LOW
SMBM_nWAIT	Tie HIGH
CFG_DATAOUT	Tie LOW

Configuration file settings

If the design does not implement the *Serial Configuration Controller* (SCC), you must set the variable FPGA_SCC to FALSE in the board `config.txt` file.

If the design does not implement the MCC-SMC interface, you must set the variable FPGA_SMB to FALSE in the board `config.txt` file.

See *3.5.2 config.txt generic board configuration file* on page 3-64.

Chapter 3
Configuration

This chapter describes the powerup and configuration processes of the MPS3 board.

It contains the following sections:

3.1 Overview of the configuration system

The *Motherboard Configuration Controller* (MCC) controls the configuration process of the MPS3 board during powerup or reset. After application of power, and a press of one of the On/Off soft reset buttons, PBON, the configuration process begins and completes without further user intervention.

Overview of configuration system

The following figure shows the board configuration system.

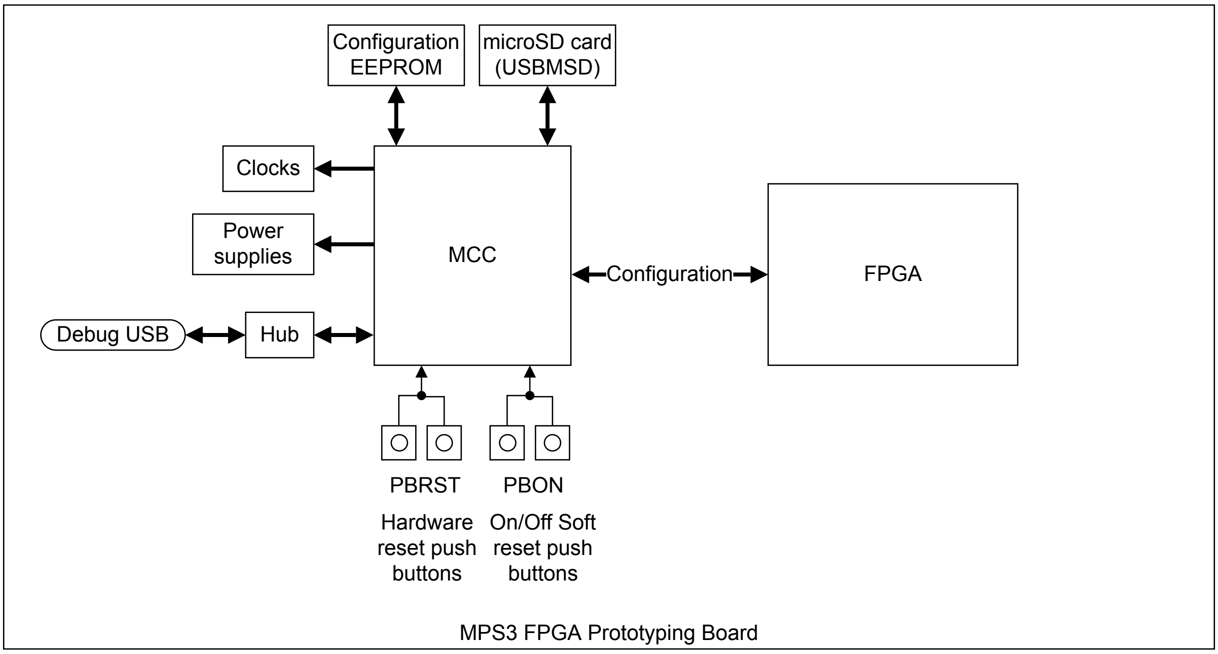


Figure 3-1 MPS3 board configuration system

The microSD card stores the board configuration files, including the `board.txt` and `config.txt` files. You can access the configuration microSD card as a *Universal Serial Bus Mass Storage Device* (USBMSD).

The MCC:

- Reads the FPGA image from the configuration microSD card and loads it into the FPGA.
- Sets the board oscillator frequencies using values from the MPS3 board configuration application note `.txt` file.
- If enabled, configures the FPGA *Serial Configuration Control* (SCC) registers using values from the `board.txt` file.
- If enabled, loads the boot memory, QSPI, DDR4, or BRAM, with the boot image that the `images.txt` file defines.

At the start of the configuration process, the MCC reads the contents of the configuration EEPROM. The EEPROM contains the following information:

- Board HBI number.
- Board revision.
- Board variant.
- Number of FPGAs.
- The names of the current images in 8.3 format and the file creation dates.

Note

The HBI number is a unique code that identifies the board. The root directories in the EEPROM and the microSD card contain subdirectories in the form HBIBoardNumberBoardrevision, for example HBI0309.

There are two stages in programming and configuring the images in the FPGA:

1. The MCC reads the configuration files in the microSD card to determine which image to load.

   Note

   The debug USB port supports MSD class enabling *Drag-and-Drop* for transferring new images to the configuration microSD card. The microSD card appears in the file system as a device with removable storage.

2. The MCC loads the FPGA image into the FPGA.

Configuration port connected to an external workstation

If you connect an external workstation to the MCC debug port, you can access the configuration microSD card. You can then edit and copy configuration and software images to the SD card.

Related information

3.5.1 Overview of configuration files and microSD card directory structure on page 3-63

3.5.2 config.txt generic board configuration file on page 3-64

3.5.3 Contents of the MB directory on page 3-64

3.5.4 Contents of the SOFTWARE directory on page 3-66

3.2 Remote USB operation

You can control the *Motherboard Configuration Controller* (MCC) through the Debug USB port.

To enable remote USB operation, you must set the USB_REMOTE parameter in the `config.txt` file to:

- TRUE to enable the remote USB feature.
- FALSE to disable the feature.

You initiate a command by putting a file into the USBMSD root directory through the USB port. You can reboot the system, reset the system, or shut down the system by using one of the following filenames:

- `reboot.txt`.
- `reset.txt`.
- `shutdown.txt`.

The MCC detects the presence of the files, performs the requested command, and deletes the file.

Note

The contents of the files have no effect. They can be empty files.

Related information

1.3 Location of components on the MPS3 board on page 1-15

A.1.6 Debug USB 2.0 connector on page Appx-A-75

3.3 Powerup and configuration sequence

The power push buttons and configuration files control the sequence of events of the board powerup and configuration process.

The following figure shows the powerup and configuration sequence.

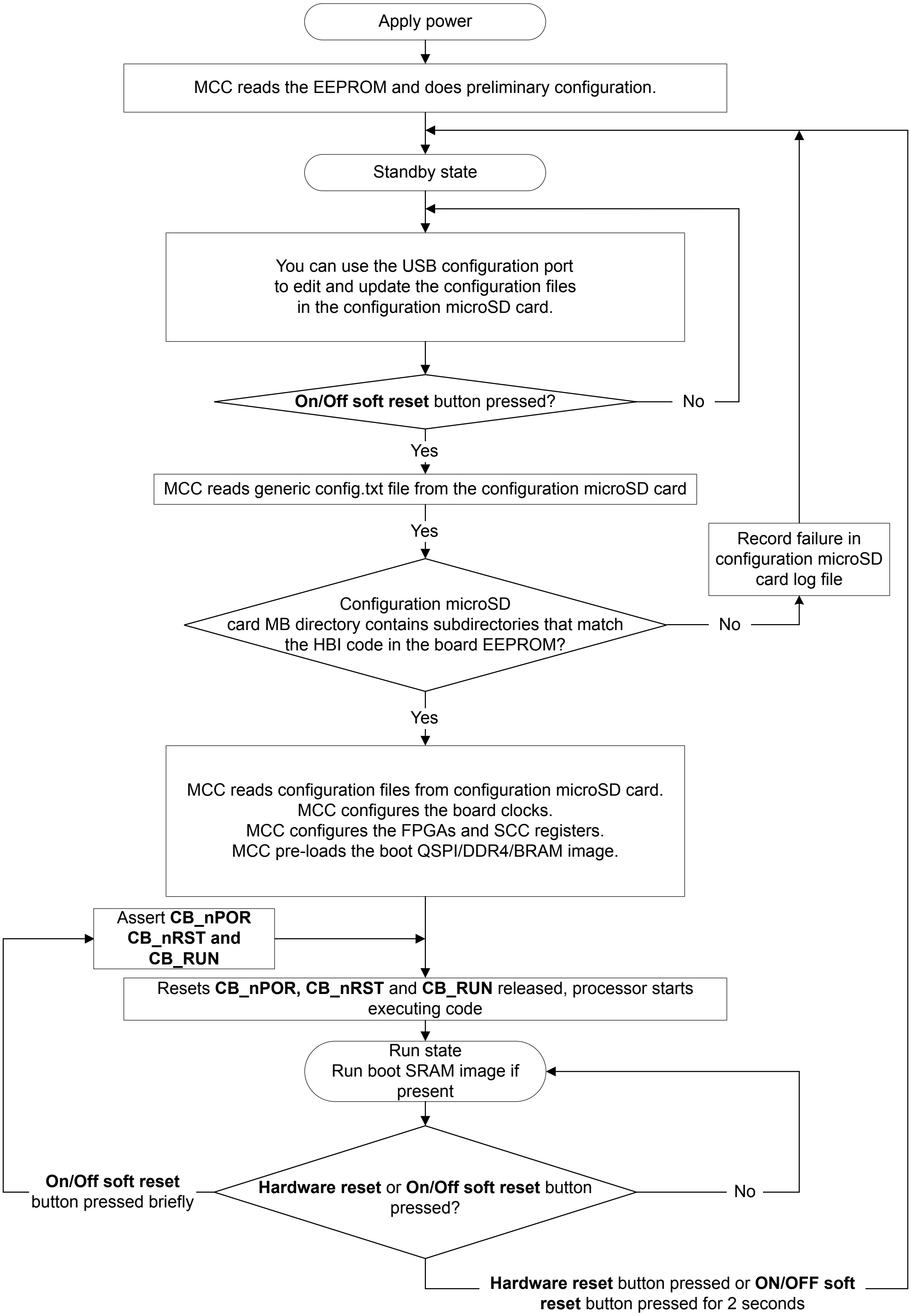

Apply power
MCC reads the EEPROM and does preliminary configuration.
Standby state
You can use the USB configuration port to edit and update the configuration files in the configuration microSD card.
On/Off soft reset button pressed?
No
Yes
MCC reads generic config.txt file from the configuration microSD card
Yes
Configuration microSD card MB directory contains subdirectories that match the HBI code in the board EEPROM?
No
Record failure in configuration microSD card log file
Yes
MCC reads configuration files from configuration microSD card.
MCC configures the board clocks.
MCC configures the FPGAs and SCC registers.
MCC pre-loads the boot QSPI/DDR4/BRAM image.
Assert **CB_nPOR CB_nRST and CB_RUN**
Resets **CB_nPOR, CB_nRST** and **CB_RUN** released, processor starts executing code
Run state
Run boot SRAM image if present
Hardware reset or **On/Off soft reset** button pressed?
On/Off soft reset button pressed briefly
No
Hardware reset button pressed or **ON/OFF soft reset** button pressed for 2 seconds

Figure 3-2 MPS3 board powerup and configuration sequence

The powerup and configuration sequence is:

1. The board applies power to the system.
2. The MCC powers the EEPROM and reads it to determine the HBI identification code for the board.
3. The system enters standby state.
4. The system enables the microSD memory card. You can connect a workstation to the debug USB port to edit existing configuration files or *Drag-and-Drop* new configuration files.

   Caution

   File names and directory names are in 8.3 format:

   - File names that you generate must be in lowercase.
   - Directory names must be in uppercase.
   - All configuration files must use DOS line endings (`0x0D`/`0x0A`).

5. The system stays in standby state until you press the On/Off soft reset button, *PBON*.
6. The system loads the board configuration file:
   - The MCC reads the generic `config.txt` file.
   - The MCC searches the microSD card `MB` directory for the `HBI0309x` subdirectory that matches the HBI code in the board EEPROM.
7. The next steps depend on the configuration files:
   - If the MCC finds configuration subdirectories that match the HBI code of the board, configuration continues and the MCC reads the `board.txt` file.
   - If the MCC does not find the correct configuration files, it records the failure in a log file on the microSD card. Configuration stops and the system re-enters the standby state.
8. The MCC measures the board power supplies.
9. The MCC configures the board clocks and FPGA SCC registers.
10. If the MCC finds new software images, it loads them into the QSPI flash, BRAM, or DDR4 memory through the FPGA.
11. The MCC releases the system resets, **CB_nPOR**, **CB_nRST**, and **CB_RUN**. The system enters the run state.
12. Normal operation continues until a new event occurs.

Related information

3.5.1 Overview of configuration files and microSD card directory structure on page 3-63

3.5.2 config.txt generic board configuration file on page 3-64

3.5.3 Contents of the MB directory on page 3-64

3.5.4 Contents of the SOFTWARE directory on page 3-66

3.4 Reset push buttons

The MPS3 board provides push buttons that initiate reset and configuration. You can initiate a software reset, or a hardware reset, of the system.

Software reset push buttons

The MPS3 board provides two On/Off soft reset buttons, both labeled *PBON*. Pressing either one initiates a software reset of the system because the two buttons perform the same function:

White On/Off soft reset button
Located near to the configuration microSD card.

Red On/Off soft reset button
Located near the debug USB 2.0 port.

See *1.3 Location of components on the MPS3 board* on page 1-15 for the location of the On/Off soft reset push buttons on the MPS3 board.

Initiate a software reset by briefly pressing the button during runtime. The MCC performs a software reset of the FPGA and resets the devices on the board.

The software reset sequence is as follows:

1. Briefly press one of the two *PBON* buttons.

   Caution

   Pressing and holding one of the software reset buttons for more than two seconds performs a hardware reset and the system enters the standby state. Pressing one of the software reset buttons for more than two seconds is similar to pressing the hardware reset button *PBRST*.

2. The MCC asserts the CB_nRST signal. The MCC might also assert **CB_nPOR** depending on the variable `ASSERTNPOR` in the configuration file `config.txt`.
3. The MCC releases **CB_nPOR** if it is active depending on the setting of the variable `ASSERTNPOR` in the configuration file `config.txt`.
4. The MCC releases **CB_nRST**.
5. The board enters the run state.

Note

The MCC does not perform the following actions as a result of a software reset:

- Read the configuration files.
- Perform a board reconfiguration.
- Perform a full reconfiguration of the FPGA.

Hardware reset buttons

The MPS3 board provides two hardware reset buttons, both labeled *PBRST*. Pressing either one initiates a hardware reset of the system because the two buttons perform the same function.

White hardware reset button
Located near to the configuration microSD card.

Red hardware reset button
Located near the debug USB 2.0 port.

See *1.3 Location of components on the MPS3 board* on page 1-15 for the location of the hardware reset push buttons on the MPS3 board.

You can change the operation of the board from ON to standby by briefly pressing one of the hardware reset buttons. Briefly pressing the button switches off the power to the board and resets the system to the default values.

If you then press one of the software reset buttons, the system performs a full configuration and enters the run state.

Related information

1.3 Location of components on the MPS3 board on page 1-15

3.5 Configuration files

This section describes the MPS3 board configuration files on the configuration microSD card that control the board powerup and configuration process.

This section contains the following subsections:

- *3.5.1 Overview of configuration files and microSD card directory structure* on page 3-63.
- *3.5.2 config.txt generic board configuration file* on page 3-64.
- *3.5.3 Contents of the MB directory* on page 3-64.
- *3.5.4 Contents of the SOFTWARE directory* on page 3-66.

3.5.1 Overview of configuration files and microSD card directory structure

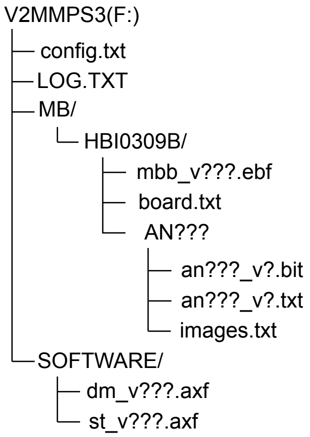
The configuration microSD card on the MPS3 board contains configuration files that control the board powerup and configuration process.

Because the board microSD card is non-volatile memory, it is only necessary to load new configuration files if you change the system configuration. The microSD card that is supplied with the MPS3 board contains default configuration files.

If you connect a workstation to the debug USB port, the microSD card appears as a *USB Mass Storage Device* (USBMSD). You can then add, edit, or delete files.

You can use a standard text editor that produces DOS line endings to read and edit the board configuration files.

The following figure shows a typical example of the directory structure on the microSD card memory.



```
V2MMPS3(F:)
 ├─ config.txt
 ├─LOG.TXT
 ├─MB/
 │   └─ HBI0309B/
 │          ├─ mbb_v???.ebf
 │          ├─ board.txt
 │          └─  AN???
 │                 ├─ an???_v?.bit
 │                 ├─ an???_v?.txt
 │                 └─ images.txt
 └─SOFTWARE/
     ├─ dm_v???.axf
     └─ st_v???.axf
```

Figure 3-3 Example USBMSD directory structure

Caution

File names and directory names are in 8.3 format:

- File names that you generate must be in lowercase.
- Directory names must be in uppercase.
- All configuration files must use DOS line endings (`0x0D`/`0x0A`).

The directory structure and file name format ensure that each image is matched to the correct target device defined in the board configuration EEPROM:

`config.txt` file Generic configuration file for all boards. This file applies to all Arm Versatile Express motherboards as well as the MPS3 board. Some settings are specific to the MPS3 board.

`MB` directory — Contains subdirectories for any board variants that might be present in the system. The subdirectory names match the HBI codes for the specific board variants. The files in this directory contain clock, register, and other settings for the board.

`SOFTWARE` directory — Can contain user application files that the MCC can load into the QSPI, BRAM, or DDR4 on the board. The memory that is loaded depends on the image that you implement in the FPGA. The `images.txt` file defines the file that the MCC loads.

Related information

3.1 Overview of the configuration system on page 3-55

3.3 Powerup and configuration sequence on page 3-58

3.5.2 config.txt generic board configuration file

You can connect a workstation to the USB debug port to update the generic configuration file, `config.txt`, on the configuration microSD card.

The following example shows a configuration file that you can load into the configuration microSD card.

Note

- Colons (:) indicate the end of commands and must be separated by a space character (`0x20`) from the value fields.
- Semicolons (;) indicate comments.

```
TITLE: Arm MPS3 FPGA prototyping board configuration file

[CONFIGURATION]
AUTORUN: FALSE              ;Auto Run from power on
AUTORUNDELAY: 3             ;Delay in seconds to wait for key press to stop bootup
RTC: FALSE                  ;TRUE = Enable RTC, FALSE = Disable RTC

UARTMODE: 2                 ;0-MCC:FPGA0, 1-MCC:FPGA1, 2-MCC/FPGA0:FPGA1
REMAP: BRAM                 ;Boot device BRAM/DDR/QSPI
FPGA_SMB: TRUE              ;FPGA image supports the MCC SMB interface
FPGA_SCC: TRUE              ;FPGA image supports the MCC SCC interface

FMC_FORCE: FALSE            ;Force FMC power ON
FMC_EEMODE: 0               ;FMC EEPROM mode, 0-default/1-two byte address

DVIMODE: VGA                ;VGA/SVGA/XGA/SXGA/UXGA or HD1080 (MCC sets OSC5)

USERSWITCH: 00000000        ;Userswitch[7:0] in binary
CONFSWITCH: 00000000        ;Configuration Switch[7:0] in binary
ASSERTNPOR: TRUE            ;External resets assert nPOR
CPU0WAIT:   0x00000000      ;CPU0WAIT value set to 0xFFFFFFFF when using CB_nRST
WDTRESET: NONE              ;Watchdog reset options NONE/RESETMB/RESETDB

USB_REMOTE: FALSE           ;Selects remote command via USB

MACADDRESS: 0xFFFFFFFFFFFF  ;MAC Address
```

Related information

3.1 Overview of the configuration system on page 3-55

3.3 Powerup and configuration sequence on page 3-58

3.5.3 Contents of the MB directory

The MPS3 board MB directory contains a configuration HBI subdirectory that matches the HBI code of the board.

The HBI subdirectory contains:

- A file of the form `mbb_vxxx.ebf`. This file is an MCC BIOS image.
- A `board.txt` file that defines the MCC BIOS image.
- An application-specific subdirectory that contains the following board configuration files:
  - Image files for the FPGA, that have `.bit` extensions, and for the MCC, that have `.rbe` extensions.

- An `images.txt` file that defines the `.axf` files that the MCC loads into external memory during configuration:
- An application note `.txt` file that defines:
  - — The number of FPGAs on the board.
  - — The number of oscillators and their frequencies.
  - — FPGA image file.
  - — Details of the SCC registers.

The following example shows a typical MPS3 board configuration `board.txt` file.

```
BOARD: HBI0309B
TITLE: Motherboard configuration file

[MCCS]
MBBIOS mbb_v132.ebf                  ;MB BIOS IMAGE

[APPLICATION NOTE]                   ;Please select the required processor
APPFILE: AN524\an524_v1.txt          ;AN524 Cortex M33 SSE200 Subsystem for MPS3
```

The following example shows a typical MPS3 board configuration application note `.txt` file.

```
BOARD: HBI0309
TITLE: AN524 application note configuration file

[FPGAS]
TOTALFPGAS: 1                        ;Total Number of FPGAS
F0FILE: an524_v1.bit                 ;FPGA0 Filename
F0MODE: FPGA                         ;FPGA0 Programming Mode

[OSCCLKS]
TOTALOSCCLKS: 7

;Clock generators OSC1 to OSC5 connect to FPGA top level signals
;OSCCLK[1] to OSCCLK[5] respectively.
;Clockgen OSC0 Drives XTAL1/CLKIN of LAN9220 Ethernet controller.
;FPGA top level signal OSCCLK[0] is driven by a fixed 24MHz reference.
;Clockgen OSC0 does not connect to FPGA top level signal OSCCLK[0].
;Clockgen OSC6 drives the DDR reference clock c0_sys_clk_p/n.

OSC0: 25.0                           ; Ethernet reference 25MHz
;     24.0                             OSCCLK[0]      - refclk
OSC1: 32.0                           ; OSCCLK[1]      - ACLK
OSC2: 50.0                           ; OSCCLK[2]      - MCLK
OSC3: 50.0                           ; OSCCLK[3]      - GPUCLK
OSC4: 24.576                         ; OSCCLK[4]      - AUDCLK
OSC5: 23.75.0                        ; OSCCLK[5]      - HDLCD (MCC overrides this value)
OSC6: 100.0                          ; GTX clock (DDR)

[HARDWARE CONTROL]
ASSERTNPOR: TRUE                     ;External resets assert nPOR
LEGACYRST: FALSE                     ;Legacy CB_nPOR/CB_nRST reset mode
CPUWAIT: 0x00000002                  ;CPUWAIT value, set to 0xFFFFFFFF when using CB_nRST

[PERIPHERAL SUPPORT]
FPGA_SMB: TRUE                       ;SMB interface is supported (MCC_SMC<>FPGA_SMB)

FPGA_SCC: TRUE                       ;SCC interface is supported
SCCREG: 0x05300000                   ;SCC registers base address

FPGA_DDR: TRUE                       ;DDR interface is supported
DDRBASE: 0x05208000                  ;DDR I2C register address

FPGA_SYSREG: TRUE                    ;System register interface is supported
FPGAREG: 0x05302000                  ;System registers base address

FPGA_REMAP: TRUE                     ;REMAP interface is supported
REMAPREG: 0x05300000                 ;REMAP register address
REMAP: BRAM                          ;REMAP boot device BRAM/DDR/QSPI
REMAPVAL: 0                          ;REMAP register value e.g. 0-BRAM. 1-QSPI

FPGA_HDMI: FALSE                     ;HDMI interface is supported
HDMIBASE: 0x05207000                 ;HDMI I2C register address

FPGA_LAN: TRUE                       ;LAN LAN9220 interface is supported
LANBASE: 0x05400000                  ;LAN LAN9220 base address

FPGA_RTC: TRUE                       ;RTC PL031 interface is supported
RTCBASE: 0x0530B000                  ;RTC PL031 base address

FPGA_QSPI: TRUE                      ;QSPI interface is supported
```

```
QSPIBASE: 0x05801000                 ;QSPI controller base address
QSPIDATA: 0x02000000                 ;QSPI data address
XIPBASE: 0x05800000                  ;QSPI XIP controller base address
QSPISCC: 0x08                        ;QSPI SCC register

[SCC REGISTERS]
TOTALSYSCONS: 1                      ;Total Number of SYSCON registers defined
SYSCON: 0x000 0x00000001             ;
```

If a particular base address is not defined in the PERIPHERALS section, the *Motherboard Configuration Controller* (MCC) uses the address in *Application Note AN524 Example SSE-200 Subsystem for MPS3*. If you use another image which does not implement a particular peripheral, you must set the base address to `0x0`. The `0x0` base address prevents the MCC from attempting to configure that peripheral.

The following example shows a typical MPS3 board `images.txt` file.

```
TITLE: Arm MPS3 FPGA prototyping board Images Configuration File

[IMAGES]
TOTALIMAGES: 1                        ;Number of Images (Max : 32)

IMAGE0ADDRESS:  0x05000000            ;Please select the required executable program
IMAGE0UPDATE: AUTO                    ;Image Update:NONE/AUTO/FORCE
IMAGE0FILE: \SOFTWARE\st_viot.axf     ; - selftest uSD
;IMAGE0FILE: \SOFTWARE\st_emmc.axf    ; - selftest emmc, also supports QSPI boot
;IMAGE0DILE: \SOFTWARE\demo_IoT.axf   ; - V2M-MSP3 demo
;IMAGE0FILE: \SOFTWARE\shield.axf     ; - Shield demo
```

See *2.7 MCC-SMC interface* on page 2-30 for information on the IMAGE0ADDRESS variable.

`.axf` and `.elf` files are treated as elf files. All other files are treated as binary.

Related information

3.1 Overview of the configuration system on page 3-55

3.3 Powerup and configuration sequence on page 3-58

3.5.4 Contents of the SOFTWARE directory

The `SOFTWARE` directory can contain applications that you can load into the QSPI flash, BRAM, or DDR4 memory. The memory that is loaded depends on the image that you implement in the FPGA.

You can create applications and load them into the memory on the board. Application images are typically boot images or demo programs and have a `.axf` extension.

Typical applications in this directory are:

- `demo_IoT.axf` board demonstration software.
- `st_viot.axf` board test software.

Related information

3.1 Overview of the configuration system on page 3-55

3.3 Powerup and configuration sequence on page 3-58

3.6 MCC command-line interface

The MPS3 board command-line interface supports system command-line input to the MCC.

This section contains the following subsections:

3.6.1 Overview of the MCC command-line interface

You must connect a workstation to UART0 to enter system commands.

The workstation settings must be:

- 115KBaud.
- 8N1 which is 8 data bits, no parity, one stop bit.
- No hardware or software flow control.

3.6.2 MCC main command menu

The following table shows the MPS3 board MCC main menu system commands.

Table 3-1 MCC main menu system commands

Command	Description
`CAP` *`filename`*`[/A]`	Capture serial data to the file *filename*. Use the /A option to append data to an existing file.
`Copy` *`input_filename_1`* *`[input_filename_2]`* *`output_filename`*	Copy file *input_filename_1* to *output_filename*. Option *input_filename_2* merges *input_filename_1* and *input_filename_2*.
`DEBUG`	Change to the debug menu.
`DEL` *`filename`*	Delete file *filename*.
`DIR [`*`mask`*`]`	Display a list of files in the directory.
`EEPROM`	Change to the EEPROM menu. **Caution** The EEPROM menu is reserved for factory use only.
`FILL` *`filename`* `[`*`nnnn`*`]`	Create a file *filename* filled with text. nnnn specifies the number of lines to create. The default is 1000.
`FORMAT [`*`label`*`]`	Format Flash Memory Card.
`HELP` or `?`	Display this help.
`HELP configfile`	Display config file flags
`HELP boardfile`	Display board file flags
`HELP imagefile`	Display image file flags
`HELP version`	Display MCC version history

Table 3-1 MCC main menu system commands (continued)

Command	Description
`REBOOT`	Cycle system power and reboot.
`REN` *`filename_1 filename_2`*	Rename a file from *filename_1* to *filename_2*.
`RESET`	Reset the board using the **nSRST** reset signal.
`SHUTDOWN`	Shut down the power supply but leave the MCC powered. The board returns to standby state.
`TYPE` *`filename`*	Display the contents of text file *filename*.
`USB_ON`	Enable MCC USB 2.0 configuration port.
`USB_OFF`	Disable MCC USB 2.0 configuration port.
`READ_AXI` *`filename address end_address`*	Read system memory to file *filename* from *address* to *end_address*. —— **Note** —— This command: • Is only ever available in run state. • Might not be available in your particular FPGA image. ————
`WRITE_AXI` *`filename address`*	Write file *filename* to system memory at *address*. —— **Note** —— This command: • Is only ever available in run state. • Might not be available in your particular FPGA image. ————

3.6.3 MCC debug menu

To switch to the debug menu, enter DEBUG at the MCC main menu. The debug menu is valid only in the run state.

The following table shows the debug commands.

Table 3-2 MCC debug command menu

Command	Description
`CFG` *`R|W OSC|V|TEMP|SCC device`* `[`*`data`*`]`	Read|Write SPI configuration command. For SCC, device is the register address.
`CFG` *`W DVIVGA|SVGA|XGA|SXGA|UXGA`*	Write DVI configuration command.
`DATE`	Display current date.
`DEBUG` *`0|1`*	Enable or disable debug printing: 0 Disable. 1 Enable.
`DEPOSIT` *`address_data`*	Write word to system memory address.
`EXAM` *`address`* `[`*`nnnn`*`]`	Examine system memory address at *address*. nnnn is number, in hex, of words to read.

Table 3-2 MCC debug command menu (continued)

Command	Description
`EXIT` or `QUIT`	Return to main menu.
`HELP` or `?`	Display this help.
`TIME`	Display current time.

3.6.4 MCC EEPROM menu (Reserved for factory use only)

To switch to the EEPROM menu, enter EEPROM at the MCC main menu. The contents of the MPS3 board EEPROMs identify the specific board variant and might contain data to load to other devices on the board.

——— Caution ———

Do not modify the factory-programmed EEPROM values.

The following table shows the EEPROM commands.

Table 3-3 EEPROM commands

Command	Description
`CONFIG 0` *`filename`*	Write configuration file to EEPROM.
`EXIT` or `QUIT`	Return to main menu.
`ERASECON [0]`	Erase configuration section of EEPROM.
`ERASEDEV [0]`	Erase device section of EEPROM.
`ERASERANGE [0] [`*`start`*`] [`*`end`*`]`	Erase EEPROM between *start* and *end*.
`ERASEIMAGE` *`image_id`*	Erase image stored in board EEPROM.
`ERASEIMAGES`	Erase images stored in board EEPROM.
`HELP` or `?`	Display this help.
`READIMAGES`	Read images stored in board EEPROM.
`READCF [0]`	Read configuration EEPROM.
`READRANGE [0] [`*`start`*`] [`*`end`*`]`	Read EEPROM between [*start*] and [*end*].

Appendix A
Signal descriptions

This appendix lists the signals at the interface connectors of the MPS3 board.

It contains the following sections:

A.1 Debug connectors

The MPS3 board provides connectors that support P-JTAG processor debug, F-JTAG FPGA debug, 16-bit and 4-bit trace, and SWD.

This section contains the following subsections:

A.1.1 20-pin IDC connector

The MPS3 board provides one 1V8 20-pin IDC connector that supports P-JTAG processor debug to enable connection of DSTREAM, or a compatible third-party debugger. The connector also supports *Serial Wire Debug* (SWD).

The 20-pin IDC connector connects to general-purpose pins on the FPGA. The availability of P-JTAG or SWD depends on the design that you implement in the FPGA.

The following figure shows the 20-pin IDC connector, J14.

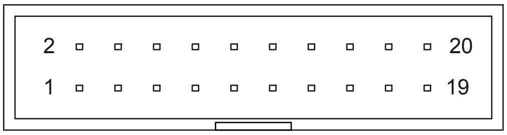


Figure A-1 20-pin IDC connector

The following table shows the pin mapping for each P-JTAG and SWD signal on the 20-pin IDC connector.

Table A-1 20-pin IDC connector, J14, pin mapping

Pin	Signal	Pin	Signal
1	**1V8_REF**	2	**1V8**
3	**nTRST**	4	**GND**
5	**TDI**	6	**GND**
7	**SWDIO/TMS**	8	**GND**
9	**SWDCLK/TCK**	10	**GND**
11	**GND/RTCK**	12	**GND**
13	**SWO/TDO**	14	**GND**
15	**nSRST**	16	**GND**
17	NC/**DBGRQ**	18	**GNDDETECT**
19	NC/**DBACK**	20	**GND**

Note

- Pins 1, 3, 5, 7, 13, 15, and 19 have pullup resistors to **1V8**.
- Pins 9, 11, and 17 have pulldown resistors to **GND**.

Related information

2.18 System debug on page 2-50

1.3 Location of components on the MPS3 board on page 1-15

A.1.2 10-pin IDC connector

The MPS3 board provides one 1V8 10-pin IDC connector that supports P-JTAG processor debug to enable connection of DSTREAM or a compatible third-party debugger. The connector also supports *Serial Wire Debug* (SWD).

The 10-pin IDC connector connects to general-purpose pins on the FPGA. The availability of P-JTAG or SWD depends on the design that you implement in the FPGA.

The following figure shows the 10-pin IDC connector, J15.

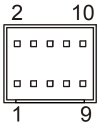


Figure A-2 10-pin IDC connector

The following table shows the pin mapping for each P-JTAG and SWD signal on the 10-pin IDC connector.

Table A-2 10-pin IDC connector, J15, pin mapping

Pin	Signal	Pin	Signal
1	**1V8**	2	**SWDIO/TMS**
3	**GND**	4	**SWDCLK/TCK**
5	**GND**	6	**SWO/TDO**
7	NC	8	NC/**TDI**
9	**GNDDETECT**	10	**nSRST**

Note

- Pins 2, 6, 8, and 10 have pullup resistors to **1V8**.
- Pin 4 has a pulldown resistor to **GND**.

Related information

2.18 System debug on page 2-50

1.3 Location of components on the MPS3 board on page 1-15

A.1.3 20-pin Cortex debug and ETM connector

The MPS3 board provides one 1V8 20-pin Cortex debug and *Embedded Trace Macrocell* (ETM) connector. The connector supports P-JTAG processor debug to enable connection of DSTREAM, or a compatible third-party debugger. The connector also supports *Serial Wire Debug* (SWD) and 4-bit trace.

The 20-pin Cortex debug and ETM connector connects to general-purpose pins on the FPGA. The availability of P-JTAG, SWD, or 4-bit trace depends on the design that you implement in the FPGA.

The following figure shows the 20-pin Cortex debug and ETM connector, J12.

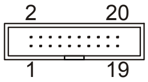


Figure A-3 20-pin Cortex debug and ETM connector

The following table shows the pin mapping for each P-JTAG, SWD, and 4-bit trace signal on the 20-pin Cortex debug and ETM connector.

Table A-3 20-pin Cortex debug and ETM connector, J12, pin mapping

Pin	Signal	Pin	Signal
1	**1V8**	2	**SWDIOTMS**
3	**GND**	4	**SWDCLKTCK**
5	**GND**	6	**SWOTDOEXTa**
7	NC	8	NC/**TDIEXTb**
9	**GNDDETECT**	10	**nSRST**
11	**3V0_OUT**	12	**TRACECLK**
13	**3V0_OUT**	14	**TRACEDATA[0]**
15	**GND**	16	**TRACEDATA[1]**
17	**GND**	18	**TRACEDATA[2]**
19	**GND**	20	**TRACEDATA[3]**

Note

- Pins 2, 6, 8, 9, 10, 11, and 13 have pullup resistors to **1V8**.
- Pin 4 has a pulldown resistor to **GND**.

Related information

2.18 System debug on page 2-50

1.3 Location of components on the MPS3 board on page 1-15

A.1.4 38-pin MICTOR connector

The MPS3 board provides one 1V8 38-pin MICTOR connector. The connector supports P-JTAG processor debug to enable connection of DSTREAM, or a compatible third-party debugger. The connector also supports *Serial Wire Debug* (SWD) and 16-bit trace.

The 38-pin MICTOR connector connects to general-purpose pins on the FPGA. The availability of P-JTAG, SWD, or 16-bit trace depends on the design that you implement in the FPGA.

The following figure shows the 38-pin MICTOR connector, J13.

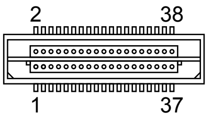


Figure A-4 38-pin MICTOR connector

The following table shows the pin mapping for each P-JTAG, SWD, and 16-bit trace signal on the 20-pin Cortex debug and ETM connector.

Table A-4 38-pin MICTOR connector, J13, pin mapping

Pin	Signal	Pin	Signal
1	NC	2	NC
3	NC	4	NC
5	**GND**	6	**TRACECLK**
7	**DBGRQ**	8	**DBGACK**
9	NC/**nSRST**	10	**EXTTRIG**
11	**TDO/SWO**	12	**1V8** reference
13	**RTCK**	14	**1V8_OUT**
15	**TCK/SWCLK**	16	**TRACEDATA[7]**
17	**TMS/SWDIO**	18	**TRACEDATA[6]**
19	**TDI**	20	**TRACEDATA[5]**
21	**nTRST**	22	**TRACEDATA[4]**
23	**TRACEDATA[15]**	24	**TRACEDATA[3]**
25	**TRACEDATA[14]**	26	**TRACEDATA[2]**
27	**TRACEDATA[13]**	28	**TRACEDATA[1]**
29	**TRACEDATA[12]**	30	**GND**
31	**TRACEDATA[11]**	32	**GNDDETECT**
33	**TRACEDATA[10]**	34	**1V8** reference
35	**TRACEDATA[9]**	36	**TRACECTL**
37	**TRACEDATA[8]**	38	**TRACEDATA[0]**

Note

- Pins 9, 11, 17, 19, and 21 have pullup resistors to **1V8**.
- Pins 13 and 15 have pulldown resistors to **GND**.

Related information

2.18 System debug on page 2-50

1.3 Location of components on the MPS3 board on page 1-15

A.1.5 14-pin F-JTAG ILA connector

The MPS3 board provides one 3V3 14-pin F-JTAG ILA connector that supports FPGA debug. It enables you to connect an ILA device, such as SignalTap II, to a hard FPGA JTAG chain in the FPGA and debug your design.

The following figure shows the 14-pin F-JTAG ILA connector, J17.

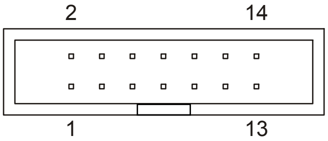


Figure A-5 14-pin F-JTAG ILA connector

The following table shows the pin mapping for each P-JTAG and SWD signal on the 14-pin F-JTAG ILA connector.

Table A-5 14-pin F-JTAG ILA connector, J17, pin mapping

Pin	Signal	Pin	Signal
1	**GND**	2	**3V3_OUT**
3	**GND**	4	**FPGA_TMS**
5	**GND**	6	**FPGA_TCK**
7	**GND**	8	**FPGA_TDO**
9	**GND**	10	**FPGA_TDI**
11	NC	12	NC
13	NC	14	NC

Note

- Pins 4, 8, and 10 have pullup resistors to **3V3**.
- Pin 6 has a pulldown resistor to **GND**.

Related information

2.18 System debug on page 2-50

1.3 Location of components on the MPS3 board on page 1-15

A.1.6 Debug USB 2.0 connector

The MPS3 board provides one USB 2.0 connector that supports configuration file editing in the microSD, UART access to the FPGA, and CMSIS-DAP FPGA debug using SWD only.

The following figure shows the USB type B connector, J8.

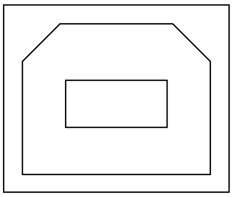


Figure A-6 Debug USB 2.0 connector

Related information

2.18 System debug on page 2-50

1.3 Location of components on the MPS3 board on page 1-15

3.2 Remote USB operation on page 3-57

A.2 Arduino Shield connectors

Connectors on the MPS3 board provide two Shield expansion interfaces. Each interface provides 16 digital I/O and six analog I/O.

The *Peripheral Module* interface (Pmod) connectors share some of the Shield connectors and are wired in parallel with them. Interface Pmod0/1 shares some signals with Shield 0 interface, and Pmod2/3 shares some signals with Shield 1 interface.

Caution

The MPS3 board supports simultaneous use of Pmod and Shield expansion but you must take care when driving the shared signals.

Shield 0 and Shield 1 interface connectors

The following figure shows a combined diagram of the Arduino Shield 0 and Arduino Shield 1 interfaces on the MPS3 board.

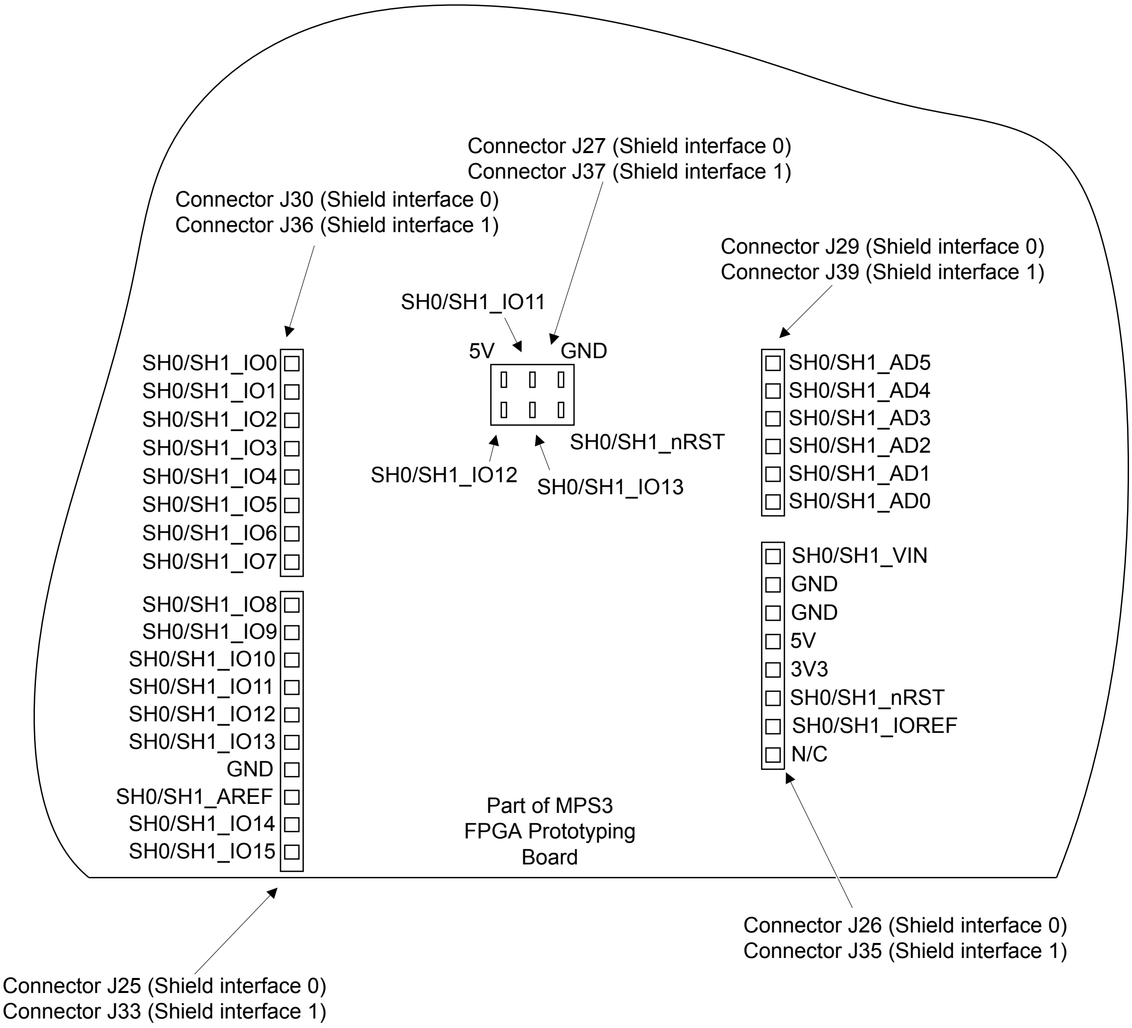


Figure A-7 Shield 0 and Shield 1 interface connectors on the MPS3 board

Connectors J25, J26, J27, J29, and J30 form the Shield 0 interface.

Connectors J33, J35, J36, J37, and J39 form the Shield 1 interface.

See *1.3 Location of components on the MPS3 board* on page 1-15 for the location of the Arduino Shield interface connectors on the MPS3 board.

Note

User-links select digital I/O operating voltages and power inputs. The power inputs and IOREF voltages have maximum current limits available at the board interface pins. See *2.16 Arduino Shield and Pmod interfaces* on page 2-43 for information on the user-links, and the maximum available IOREF currents.

Digital I/O connectors: Connectors J25, J33, J30, and J39

Connector J25 provides Shield 0 I/O[15:8], and connector J33 provides Shield 1 I/O[15:8]. The connectors also provide the analog I/O reference voltages. The following table shows the pin mapping for connectors J25 and J33.

Table A-6 Connectors J25 (Shield 0) and J33 (Shield 1) signal list

Pin	Signal
1	**SH0/SH1_IO8**
2	**SH0/SH1_IO9**
3	**SH0/SH1_IO10**
4	**SH0/SH1_IO11**
5	**SH0/SH1_IO12**
6	**SH0/SH1_IO13**
7	**GND**
8	**SH0/SH1_AREF**
9	**SH0/SH1_IO14**
10	**SH0/SH1_IO15**

Caution

FPGA pins SH0_IO16, SH0IO17, SH1_IO16, and SH1_IO17, which are unused, connect to the Shield connectors through 4K7 resistors.

- FPGA pin SH0_IO16 connects to Shield pin SH0_IO14.
- FPGA pin SH0_IO17 connects to Shield pin SH0_IO15.
- FPGA pin SH1_IO16 connects to Shield pin SH1_IO14.
- FPGA pin SH1_IO17 connects to Shield pin SH1_IO15.

Arm recommends that you set FPGA pins SH[1:0]_IO[17:16] to high impedance to prevent any possibility of interference with connector pins SH[1:0]_IO[15:14].

Connector J30 provides Shield 0 I/O[7:0] and connector J36 provides Shield 1 I/O[7:0]. The following table shows the pin mapping for connectors J30 and J36.

Table A-7 Connectors J30 (Shield 0) and J36 (Shield 1) signal list

Pin	Signal
1	**SH0/SH1_IO0**
2	**SH0/SH1_IO1**
3	**SH0/SH1_IO2**

Table A-7 Connectors J30 (Shield 0) and J36 (Shield 1) signal list (continued)

Pin	Signal
4	**SH0/SH1_IO3**
5	**SH0/SH1_IO4**
6	**SH0/SH1_IO5**
7	**SH0/SH1_IO6**
8	**SH0/SH1_IO7**

Analog I/O connectors: Connectors J29 and J39

Connector J29 provides six analog I/O for Shield 0 and connector J36 provides six analog I/O for Shield 1. The following table shows the pin mapping for connectors J29 and J39.

Table A-8 Connectors J29 (Shield 0) and J39 (Shield 1) signal list

Pin	Signal
1	**SHO/SH1_AD0**
2	**SH0/SH1_AD1**
3	**SH0/SH1_AD2**
4	**SH0/SH1_AD3**
5	**SH0/SH1_AD4**
6	**SH0/SH1_AD5**

Power and voltage references: Connectors J26 and J35

Connector J26 provides power and voltage references for Shield 0 digital I/O. Connector J35 provides power and voltage references for Shield 1 digital I/O. The following table shows the pin mapping for Shield connectors J26 and J35.

Table A-9 Connectors J26 (Shield 0) and J35 (Shield 1) signal list

Pin	Signal
1	N/C
2	**SHO/SH1_IOREF**
3	**SHO/SH1_nRST**
4	**3V3**
5	**5V**
6	**GND**
7	**GND**
8	**SHO/SH1_VIN**

Supplementary connectors J27 and J37

The supplementary connectors, J27 and J37, provide subsets of the signals on the main Shield connectors. Connector J27 provides a subset of the Shield 0 signals and connector J37 provides a subset of the Shield 1 signals. The following table shows the pin mapping for connectors J27 and J37.

Table A-10 Connectors J27 (Shield 0) and J37 (Shield 1) signal list

Pin	Signal
1	**SH0/SH1__IO12**
2	**5V**
3	**SH0/SH1_IO13**
4	**SH0/SH1_IO11**
5	**SHO/SH1_nRST**
6	**GND**

Related information

2.16 Arduino Shield and Pmod interfaces on page 2-43

1.3 Location of components on the MPS3 board on page 1-15

A.3 Peripheral Module (Pmod) connectors

The Pmod connectors on the MPS3 board provide digital I/O expansion capability, an alternative to the Shield interfaces.

The four Pmod connectors enable fitting of TYPE 2A (J24/J34) and TYPE 1 (J28/J38) boards. Connectors J24 and J28 form interface Pmod0/1 and connectors J34 and J38 form interface Pmod2/3. The Pmod connectors carry a subset of the signals on the Arduino connectors and are wired in parallel with them. Interface Pmod0/1 shares signals with Shield 0 interface, and Pmod2/3 shares signals with Shield 1 interface.

Note

User-links select 3V3 or 5V digital I/O operating voltages, and the digital IOREF voltages. The IOREF voltages have maximum current limits available at the board interface pins. See *2.16 Arduino Shield and Pmod interfaces* on page 2-43 for information on the user-links, and the maximum available IOREF current.

Caution

The MPS3 board supports simultaneous use of Pmod and Shield expansion but you must exercise caution when driving the shared signals.

The following figure shows the Pmod connectors.

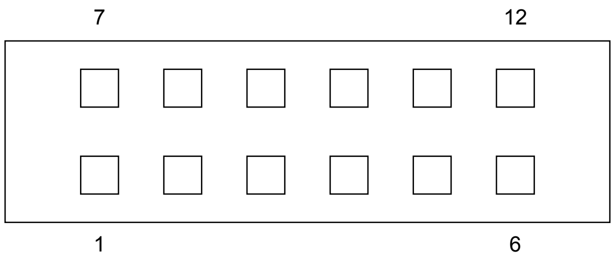


Figure A-8 Pmod connectors J24, J28, J34, and J38

The following table shows the pin mapping for the Pmod0/1 interface connector, J24.

Table A-11 Connector J24 (Pmod0/1 interface) signal list

Pin	Signal	Pin	Signal
1	**SH0_5V_IO10**	7	**SH0_5V_IO5**
2	**SH0_5V_IO11**	8	**SH0_5V_IO6**
3	**SH0_5V_IO12**	9	**SH0_5V_IO7**
4	**SH0_5V_IO13**	10	**SH0_5V_IO8**
5	**GND**	11	**GND**
6	**SHO_REF**	12	**SHO_REF**

The following table shows the pin mapping for the Pmod0/1 interface connector, J28.

Table A-12 Connector J28 (Pmod0/1 interface) signal list

Pin	Signal	Pin	Signal
1	**SH0_5V_IO3**	7	**SH0_5V_IO4**
2	**SH0_5V_IO1**	8	**SH0_5V_IO2**
3	**SH0_5V_IO0**	9	**SH0_5V_IO15**
4	**SH0_5V_IO9**	10	**SH0_5V_IO14**
5	**GND**	11	**GND**
6	**SHO_REF**	12	**SHO_REF**

The following table shows the pin mapping for the Pmod2/3 interface connector, J34.

Table A-13 Connectors J34 (Pmod2/3 interface) signal list

Pin	Signal	Pin	Signal
1	**SH0_5V_IO10**	7	**SH1_5V_IO5**
2	**SH0_5V_IO11**	8	**SH1_5V_IO6**
3	**SH0_5V_IO12**	9	**SH1_5V_IO7**
4	**SH0_5V_IO13**	10	**SH1_5V_IO8**
5	**GND**	11	**GND**
6	**SH1_REF**	12	**SH1_REF**

The following table shows the pin mapping for the Pmod2/3 interface connector, J38.

Table A-14 Connectors J38 (Pmod2/3 interface) signal list

Pin	Signal	Pin	Signal
1	**SH0_5V_IO3**	7	**SH1_5V_IO4**
2	**SH0_5V_IO9**	8	**SH1_5V_IO2**
3	**SH0_5V_IO0**	9	**SH1_5V_IO15**
4	**SH0_5V_IO1**	10	**SH1_5V_IO14**
5	**GND**	11	**GND**
6	**SH1_REF**	12	**SH1_REF**

Related information

2.16 Arduino Shield and Pmod interfaces on page 2-43

1.3 Location of components on the MPS3 board on page 1-15

A.4 FMC-HPC connector

The MPS3 board provides a 400-way Samtec SEARAY connector to support the FPGA Mezzanine Card standard, high pin count variant, FMC-HPC.

The following figure shows the FMC-HPC connector, J9.

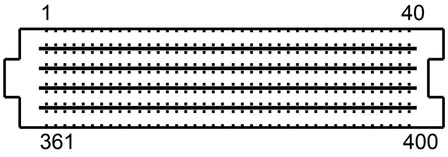


Figure A-9 FMC-HPC connector

The pin mapping of the FMC-HPC connector conforms to the *FPGA Mezzanine Card* (FMC) Standard ANSI/VITA 57.1. See *http://www.vita.com/* for the pin mapping of the FMC-HPC connector.

Arm supplies two spreadsheets that describe the FPGA-FMC connectivity on the MPS3 board. See *FPGA-FMC pin connectivity* on page 2-46 for information on how to download the spreadsheets.

Related information

2.17 FMC-HPC interface on page 2-46

1.3 Location of components on the MPS3 board on page 1-15

A.5 FMC configuration connector

The MPS3 board provides a custom 14-pin connector to enable configuration of Arm FMC boards.

The FMC configuration connector is reserved for Arm FMC boards only.

The following figure shows the FMC-HPC configuration connector, J61.

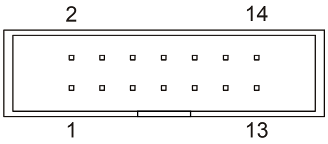


Figure A-10 FMC-HPC configuration connector

Related information

2.17 FMC-HPC interface on page 2-46

1.3 Location of components on the MPS3 board on page 1-15

A.6 Combined Ethernet and dual USB-A connector

The MPS3 board provides a combined Ethernet and dual USB-A connector that connects to the Ethernet 10/100 controller and to the USB 2.0 controller.

The following figure shows the combined Ethernet and dual USB-A connector, J2.

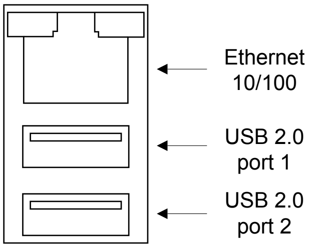


Figure A-11 Combined Ethernet and dual USB-A connector

Related information

2.8 USB 2.0 and Ethernet static memory interface on page 2-35

1.3 Location of components on the MPS3 board on page 1-15

A.7 HDMI Type A female connector

The female HDMI connector on the MPS3 board provides digital video and digital audio to external displays.

The following figure shows the HDMI connector, J3.

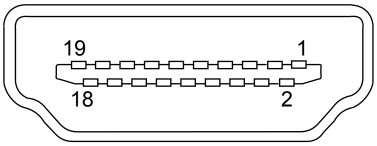


Figure A-12 HDMI connector

The following table shows the pin mapping for the HDMI connector, J3, including encoded I^2S digital audio.

Table A-15 HDMI connector, J3, signal list

Pin	Signal	Pin	Signal
1	**DVI_TX2P**	2	**GND**
3	**DVI_TX2N**	4	**DVI_TX1P**
5	**GND**	6	**DVI_TX1N**
7	**DVI_TX0P**	8	**GND**
9	**DVI_TX0N**	10	**DVI_TXCP**
11	**GND**	12	**DVI_TXCN**
13	**DVI_CECAO**	14	No connection
15	**DVI_DSCLO**	16	**DVI_DSDAO**
17	**GND**	18	**DVI_5V0**
19	**DVI_HPDO**	-	-

Related information

2.9 Video HDLCD interface on page 2-36

1.3 Location of components on the MPS3 board on page 1-15

A.8 Audio connectors, stacked stereo jacks

The MPS3 board provides three stacked 3.5mm stereo jack connectors that connect to a stereo audio codec. The connectors provide line-level stereo input, line-level stereo output, and microphone-level stereo input.

The top, blue, jack is the line-in connector. The middle, green, jack is the line-out connector. The bottom, pink, jack is the microphone-in connector.

When using electret microphones, use jumpers J58 (L) and J59 (R) to enable microphone bias current.

The following figure shows the three stereo jack connectors, J4.

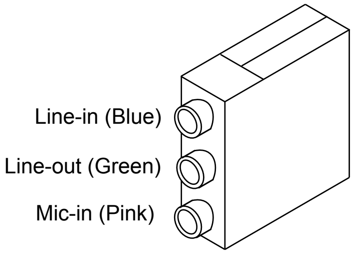


Figure A-13 Stacked stereo jack connectors

Related information

2.10 Audio codec interface on page 2-37

1.3 Location of components on the MPS3 board on page 1-15

A.9 12V power connector

The MPS3 board provides a Thru-hole DC power jack for connecting external power to the board.

Connect the external mains power supply unit, that Arm supplies with the MPS3 board, to the power jack.

Alternatively, you can connect an external 12V 5A, +/-10%, power supply to the power jack.

Note

The center pin is the positive side of the power supply.

Related information

2.5 Power on page 2-27

1.3 Location of components on the MPS3 board on page 1-15

Appendix B
Specifications

This appendix contains electrical specifications of the MPS3 board.

It contains the following section:

B.1 Available power for expansion boards

The MPS3 board supplies power to the expansion boards through the expansion connectors.

FMC-HPC power

The MPS3 board supplies power, through the FMC-HPC connector, to a fitted FMC-HPC board. The following table shows the maximum current that the board can supply from each power rail.

Table B-1 FMC-HPC power

Power rail	Voltage	Max load	Comment
12P0V	12V	1A	-
3P3VAUX	3V3 standby	20mA	For FMC EEPROM only
3P3V	3V3	2.5A	-
FMCVADJ	1V2/1V5/1V8	4A	Other voltages are not supported

Shield expansion and Pmod expansion power

The MPS3 board supplies power to expansion Shields or Pmod expansion boards, depending on what expansion boards are fitted. The following table shows the maximum current that the board can supply from each power rail.

Table B-2 Shield and Pmod power

Power rail	Max load	Comment
3V3	1A	Maximum current available for both Shields, or all four Pmod expansion boards. Includes digital I/O reference IOREF.
5V	1A	Maximum current available for both Shields, or all four Pmod expansion boards. Includes digital I/O reference IOREF.
12V	0.5A	Maximum current available for both Shields, or all four Pmod expansion boards. Includes digital I/O reference IOREF.

Note

User-links on the board select 3V3 or 5V digital I/O operation. See *2.16 Arduino Shield and Pmod interfaces* on page 2-43 for information on the digital I/O user-links.

Appendix C
Revisions

This appendix describes the technical changes between released issues of this book.

It contains the following section:

C.1 Revisions

The following table lists the technical changes between released issues of this book.

Table C-1 Issue 100765_0000_00

Change	Location	Affects
No changes, first release.	-	-

Table C-2 Differences between issue 100765_0000_00 and issue 100765_0000_01

Change	Location	Affects
Removed mention of JTAG mode from description of MPS3 DAP-Link interface.	*2.18 System debug* on page 2-50 *A.1.6 Debug USB 2.0 connector* on page Appx-A-75	All versions
Removed statement that availability of P-JTAG or SWD depends on the design that you implement in the FPGA.	*A.1.5 14-pin F-JTAG ILA connector* on page Appx-A-74	All versions

Table C-3 Differences between issue 100765_0000_01 and issue 100765_0000_02

Change	Location	Affects
Updated document title to match product name change. Updated text and diagrams to match change of document title and product name change.	Throughout document.	All versions
Added information about remote USB operation.	*3.2 Remote USB operation* on page 3-57 *2.18 System debug* on page 2-50	All versions
Updated example configuration application note `.txt` file.	*3.5.3 Contents of the MB directory* on page 3-64	All versions
Added information about command-line interface.	*3.6 MCC command-line interface* on page 3-67	All versions

Table C-4 Differences between issue 100765_0000_02 and issue 100765_0000_03

Change	Location	Affects
Changed Application Note reference from AN522 to AN524.	*Additional reading* on page 8 *2.2 Example Cortex®-M33 IoT Kit subsystem design* on page 2-21 *3.5.3 Contents of the MB directory* on page 3-64	All versions
Updated example `board.txt`. and application note `.txt` files.	*3.5.3 Contents of the MB directory* on page 3-64	All versions
Added Caution about high-value resistors between unused FPGA pins and connectors J25 and J33, underneath connector pin list table.	*A.2 Arduino Shield connectors* on page Appx-A-76	All versions

Table C-5 Differences between issue 100765_0000_03 and issue 100765_0000_04

Change	Location	Affects
Updated OSCCLKS section of example configuration application note `.txt` file.	*3.5.3 Contents of the MB directory* on page 3-64	All versions
Updated clock description and table. Renamed OSCCLK sources as OSC.	*2.3 Clocks* on page 2-23 Several figures in the document.	All versions
Added new HELP commands to MCC main menu system commands table.	*3.6.2 MCC main command menu* on page 3-67	Firmware version 3.1
Added information about Application Note AN533 download. Download includes detailed information about FPGA-FMC pin connectivity.	*FPGA-FMC pin connectivity* on page 2-46	All versions
Added link to section *FPGA-FMC pin connectivity* on page 2-46 in FMC-HPC connector description.	*A.4 FMC-HPC connector* on page Appx-A-82	All versions
Added *Application Note AN533 Blinky example FPGA image for the MPS3 Prototyping Board* to Additional Reading list.	*Additional reading* on page 8	All versions
Updated CE Conformance Notice.	*Conformance Notices* on page 3	All versions