Arm® CoreSight™ DAP-Lite2

Revision: r2p1

Technical Reference Manual

Non-Confidential

Issue 01

100572_0201_01_en

Arm® CoreSight™ DAP-Lite2

Technical Reference Manual

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-00	16 September 2016	Non-Confidential	First release for r0p0 REL
0100-00	18 June 2019	Non-Confidential	First release for r1p0 REL
0101-00	13 December 2019	Non-Confidential	First release for r1p1 REL
0200-00	27 March 2020	Non-Confidential	First release for r2p0 REL
0201-01	28 January 2022	Non-Confidential	First release for r2p1 REL

Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

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

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

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

Contents

1 Introduction

1.1 Product revision status

The `r`*`x`*`p`*`y`* identifier indicates the revision status of the product described in this manual, for example, `r`*`1`*`p`*`2`*, where:

r*x* Identifies the major revision of the product, for example, r1.

p*y* Identifies the minor revision or modification status of the product, for example, p2.

1.2 Intended audience

This book is written for the following audiences:

- Hardware and software engineers who want to incorporate CoreSight™ DAP-Lite2 into their design and produce real-time instruction and data trace information from a SoC.
- Software engineers writing tools to use CoreSight™ DAP-Lite2.

This book assumes that readers are familiar with AMBA® bus design and JTAG methodology.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

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

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

Typographic conventions

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Signal names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.

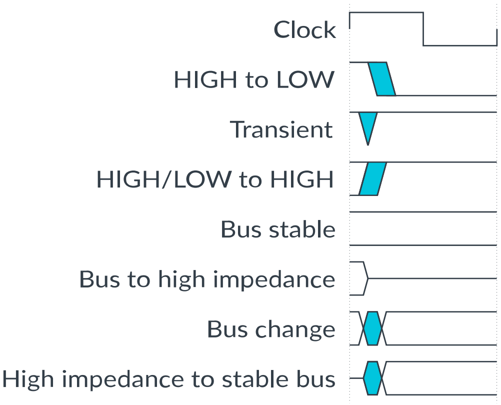
Convention	Use
`monospace bold`	Language keywords when used outside example code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.
⚠ Caution	Recommendations. Not following these recommendations might lead to system failure or damage.
⚠ Warning	Requirements for the system. Not following these requirements might result in system failure or damage.
⚠ Danger	Requirements for the system. Not following these requirements will result in system failure or damage.
Note	An important piece of information that needs your attention.
Tip	A useful tip that might make it easier, better or faster to perform a task.
Remember	A reminder of something important that relates to the information you are reading.

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 1-1: 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.

1.4 Additional reading

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

Table 1-2: Arm publications

Document name	Document ID	Licensee only
Arm® AMBA® Low Power Interface Specification	IHI 0068	No
Arm® AMBA® 5 AHB Protocol Specification AHB5, AHB-Lite	IHI 0033	No
Arm® CoreLink™ LPD-500 Low Power Distributor Technical Reference Manual	100361	No
Arm® CoreSight™ Architecture Specification v3.0	IHI 0029	No
Arm® CoreSight™ Base System Architecture Specification v1.0	DEN 0068	No
Arm® Debug Interface Architecture Specification ADIv6.0	IHI 0074	No
Arm® Power Control System Architecture Specification Version 1.0	DEN 0050	Yes

Table 1-3: Other publications

Organization	Document name
Accellera	*IP-XACT version 1685-2009*
IEEE	*Verilog-2001 Standard* IEEE Std 1364-2001

Note

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at http://www.adobe.com

2 About the DAP-Lite2

This chapter introduces the CoreSight™ DAP-Lite2.

2.1 About DAP-Lite2

CoreSight™ DAP-Lite2 enables an off-chip debugger to connect to a target system using a low pin-count JTAG or Serial Wire interface. The debugger can then control the target processor during a debug session.

DAP-Lite2 provides two *Debug Access Ports* (DAPs) that are compliant with *Arm® Debug Interface Architecture Specification ADIv6.0*. One supports application and real-time processors with AMBA APB4 debug interfaces. The other supports microcontroller processors with AMBA AHB5 debug interfaces.

DAP-Lite2 is designed for use in SoC designs that comprise single, or MPCore, Arm® Cortex®-based processors.

Figure 2-1: DAP-Lite2 Cortex-A and Cortex-R system diagram

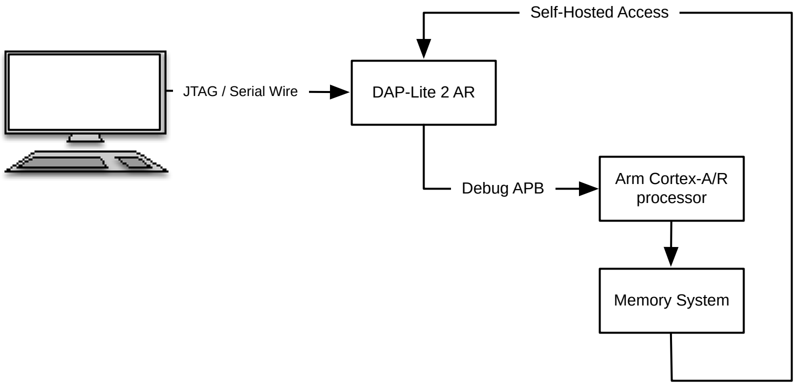
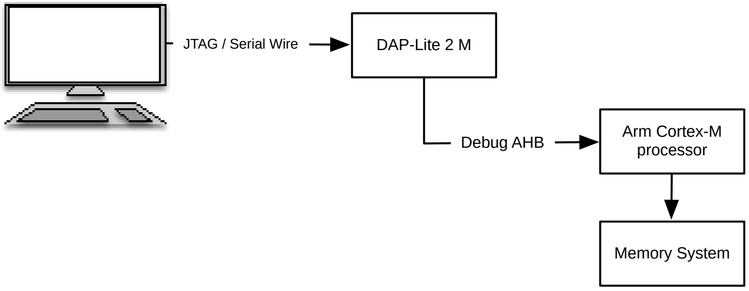


Figure 2-2: DAP-Lite2 Cortex-M system diagram



Note

DAP-Lite2 does not support the capture and export of trace data. If you require trace, contact Arm® to ask about alternative CoreSight™ products that provide such support.

DAP-Lite2 is built on Arm® CoreSight SoC-600 IP and provides the following features:

- Support for the *Arm Debug Interface* (ADI) v6 and CoreSight v3 architectures that enable you to build debug functionality into your systems
- Q-Channel interfaces for clock and power quiescence
- Arm® CoreLink LPD-500 can be integrated with DAP-Lite2 as part of a full-chip power and clock control methodology.

The DAP-Lite2 package includes:

- Two DAP components that are written in Verilog, and that are compliant with the Verilog-2001 Standard (IEEE Std 1364-2001)
- A configuration flow that:
  - Validates your component configuration choices
  - Generates IP-XACT descriptions for your chosen component configuration
  - Copies all required design files into your target directory
  - Generates UPF constraint files at component level for signals that are able to cross power domain boundaries
- Example timing constraint files for each component in SDC format
- Verification IP to assist verification of the integration of DAP-Lite2 with Arm® Cortex®-based processors

Note

If you are building a system that implements dynamic debug authorization, contact Arm to ask about CoreSight products that provide architected support for introducing certificates to the system.

2.2 Supported standards

CoreSight™ DAP-Lite2 is compliant with the following standards.

- *Arm® CoreSight™ Architecture Specification v3.0*
- *AMBA® APB Protocol Specification Version 2.0*
- *Arm® Debug Interface Architecture Specification ADIv6.0*
- *Arm® AMBA® 5 AHB Protocol Specification AHB5, AHB-Lite*
- *Arm®AMBA® Low Power Interface Specification*.

- *Verilog-2001 Standard*
- Accellera, *IP-XACT version 1685-2009*
- *IEEE 1149.1-2001 IEEE Standard Test Access Port and Boundary Scan Architecture (JTAG)*

2.3 Documentation

The DAP-Lite2 documentation includes a *Technical Reference Manual* (TRM) and a *Configuration and Integration Manual* (CIM). These books relate to the DAP-Lite2 design flow.

Technical Reference Manual

The TRM describes the functionality and the effects of functional options on the behavior of the DAP-Lite2 components. It is required at all stages of the design flow. The choices that you make in the design flow can mean that some behavior that is described in the TRM is not relevant. If you are programming a device that is based on DAP-Lite2 components, then contact the integrator to determine the configuration of your device.

Configuration and Integration Manual

The CIM describes:

- How to configure the DAP-Lite2 components
- How to integrate the DAP-Lite2 components into your SoC design and how to configure system-specific Identification Registers
- How to implement the DAP-Lite2 components to produce a hard macrocell of the design. This description includes custom cell replacement, a description of the power domains, and a description of the design synthesis.

The CIM is a confidential book that is only available to licensees.

2.4 Design process

The DAP-Lite2 components are delivered as synthesizable Verilog RTL.

Before the DAP-Lite2 components can be used in a product, they must go through the following processes:

System design

Determines the necessary structure and interconnections of the DAP-Lite2 components that form the CoreSight™ debug and trace subsystem

Configuration

Defines the memory map of the system and the functional configuration of the DAP-Lite2 components

Integration

Connects the DAP-Lite2 components together, and to the SoC memory system and peripherals

Verification

Verifies that the CoreSight™ debug and trace subsystem has been correctly integrated to the processor or processors in your SoC

Implementation

Uses the Verilog RTL in an implementation flow to produce a hard macrocell

The operation of the final device depends on:

Configuration

The implementer chooses the options that affect how the RTL source files are pre-processed. These options usually include, or exclude, logic that affects one or more of the area, maximum frequency, and features of the resulting macrocell.

Software configuration

The programmer configures the CoreSight™ debug and trace subsystem by programming specific values into registers that affect the behavior of the DAP-Lite2 components.

Note

Arm recommends that you follow the guidance *Arm® CoreSight Base System Architecture v1.0* to ensure wide support across the Arm debug ecosystem.

2.5 Component list

The following tables show the components and their versions.

Table 2-1: DAP-Lite2 component list

Name	Description	Version	Revision	IP-XACT version
`daplite2_ar`	DAP-Lite2 for Cortex-A and Cortex-R processors	r2p1	-	r2p1_0
`daplite2_m`	DAP-Lite2 for Cortex-M processors	r2p1	-	r2p1_0

Table 2-2: SoC-600 components used in DAP-Lite2 modules

Name	Description	DAP-Lite2 module	Version	Revision
`css600_ahbap`	AHB Access Port	`daplite2_m`	r2p1	4
`css600_apbap`	APB Access Port	`daplite2_ar`	r1p1	3
`css600_apbasyncbridge`	APB asynchronous bridge	`daplite2_ar`	r0p3	-
`css600_apbic`	APB interconnect	`daplite2_ar`	r0p2	-
`css600_apbrom`	APB ROM table	`daplite2_ar`	r0p1	-
`css600_dp`	Debug Port	`daplite2_ar`, `daplite2_m`	r0p4	4

The following SoC-600 components are supplied to support integration of the `daplite2_ar` into your system. Use of these components is optional.

Table 2-3: daplite2_ar supporting components

Name	Description	Version	Revision	IP-XACT version
`css600_apb3toapb4adapter`	APB3 to APB4 adapter	r0p0	-	r0p0_1
`css600_apb4toapb3adapter`	APB4 to APB3 adapter	r0p0	-	r0p0_1

Note

The Revision column only applies to components that have a programmers model. In these cases, the value that is shown is for the PIDR2.REVISION field.

2.6 Product revisions

This section describes the differences in functionality between product revisions of the CoreSight DAP-Lite2.

r0p0

First release of CoreSight™ DAP-Lite2. Based on CoreSight Architecture v2.0 and ADIv5.0. Support for Arm Cortex-A and Arm Cortex-R processor families only.

r1p0

Second release of CoreSight™ DAP-Lite2. Based on CoreSight SoC-600 components, and CoreSight Architecture v3.0 and ADIv6.0. Support for Arm Cortex-A, Arm Cortex-R, and Arm Cortex-M processor families.

r1p1

Third release of CoreSight™ DAP-Lite2. DP component errata fixes.

r2p0

Fourth release of CoreSight™ DAP-Lite2. AHB-AP component enhancement to extend support of AHB access HPROT attributes.

Note

This changes the programmers model of the AHB-AP. See CSW for the extension of HPROT support. See IDR and PIDR2 for identifying these changes.

r2p1

Fifth release of CoreSight DAP-Lite2. AHB-AP and APB-AP components updated.

3 DAP-Lite2 functional description

This chapter describes the functionality of the DAP-Lite2 modules.

3.1 DAP-Lite2 for Arm Cortex-A and Arm Cortex-R processors

The `daplite2_ar` block is the top level of the DAP-Lite2 for Arm Cortex-A and Arm Cortex-R processors.

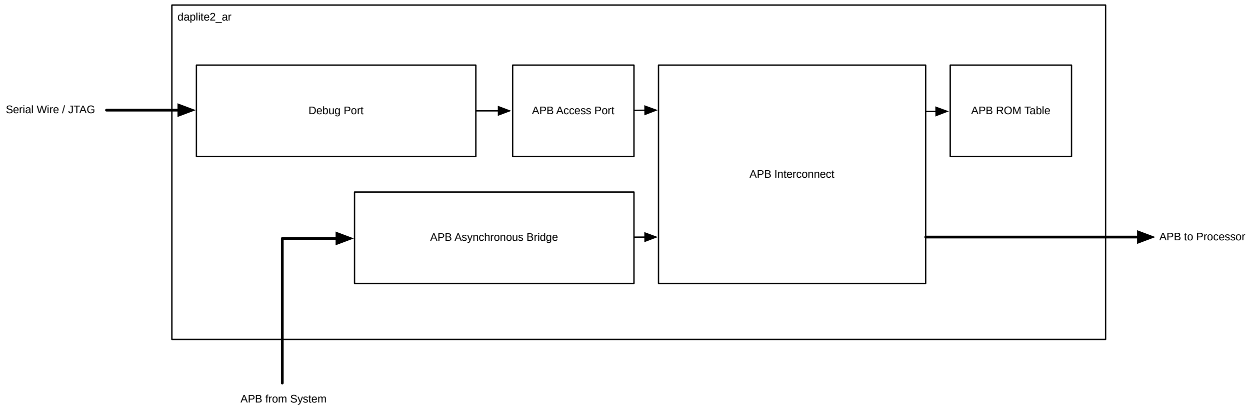
`daplite2_ar` integrates several SoC-600 components to provide the necessary signals to interface to:

- An external debugger
- An Arm Cortex-A processor or Arm Cortex-R processor
- The memory system of the processor, for self-hosted debug access
- A system-level power and reset controller
- An authentication controller

`daplite2_ar` is an integration of:

- `css600_dp` Debug Port
- `css600_apbap` APB Access Port
- `css600_apbic` APB interconnect
- `css600_apbrom` APB ROM table
- `css600_apbasynbridge` APB asynchronous bridge

Figure 3-1: DAP-Lite2 AR block diagram



See 6 SoC-600 components programmers model on page 35 for details of the components.

3.2 DAP-Lite2 for Arm Cortex-M processors

The `daplite2_m` block is the top level of the DAP-Lite2 for Arm Cortex-M processors.

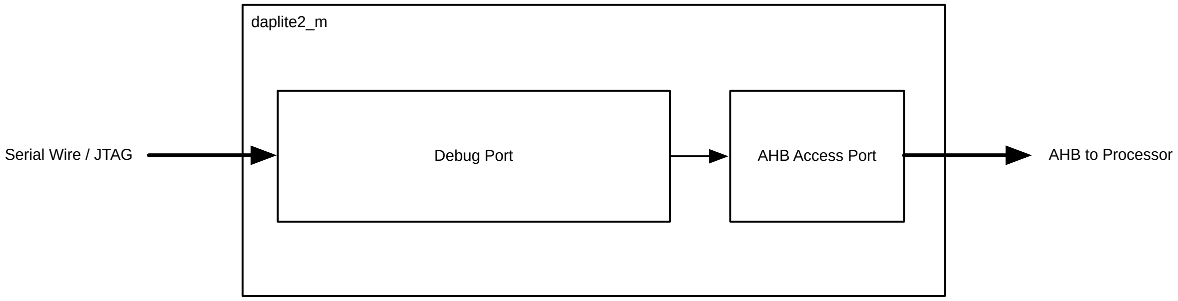
`daplite2_m` integrates several SoC-600 components to provide the necessary signals to interface to:

- An external debugger
- An Arm Cortex-M processor
- A system-level power and reset controller
- An authentication controller

`daplite2_m` is an integration of:

- `css600_dp` Debug Port
- `css600_ahbap` AHB Access Port

Figure 3-2: DAP-Lite2 M block diagram



See 6 SoC-600 components programmers model on page 35 for details of the components.

3.3 Interfaces

DAP-Lite2 has the following interfaces.

3.3.1 SWJ

The *Serial Wire JTAG* (SWJ) interface provides a low pin-count interface to an off-chip external debugger.

The SWJ interface supports both *Serial Wire Debug* (SWD) and JTAG data link protocols on a single set of shared pins, with dynamic switching between the two protocols.

The Serial Wire protocol requires two hardware pins and is targeted at pin-constrained systems.

The JTAG protocol requires four hardware pins, and allows for an extra, optional, reset signal. The JTAG interface requires more interface pins than Serial Wire. However, it can be daisy-chained with other JTAG TAPs on chip, for example, boundary scan controllers.

3.3.2 APB, daplite2_ar

The AMBA 4 APB interfaces provide connection into the debug subsystem and connection from the memory system.

Extra wakeup signals are provided. These augment standard AMBA 4 APB functionality. They enable integration into systems that support aggressive clock gating for minimization of dynamic power.

3.3.3 AHB, daplite2_m

The AMBA 5 *Advanced High-performance Bus* (AHB) interface provides connection into the debug subsystem.

Extra wakeup signals are provided. They augment standard AMBA 5 AHB functionality. They enable integration into systems that support aggressive clock gating for minimization of dynamic power.

3.3.4 Power and reset control

The Power and Reset control interfaces provide request and acknowledge signals for integration with a system power and reset controller.

In systems that support independent power control of the debug subsystem, the request and acknowledge signals permit static power to be minimized when the system is not being debugged.

3.3.5 Q-Channel LPI

The *Q-Channel Low-Power Interfaces* (LPIs) provide a standard mechanism for interfacing to a system clock and power controller.

The LPIs enable the power controller to request clock and power quiescence to minimise power consumption.

3.3.6 Access Port Enable

The Access Port enable interface provides a standard mechanism for the system security controller to grant permission to the DAP-Lite2 to perform accesses in the debug subsystem.

3.4 Clocks and resets

DAP-Lite2 implements multiple clock and reset domains for integration into the target system.

The external debugger drives the SWJ interface clock **swclktck**.

The `daplite2_ar` APB master and slave interfaces, and `daplite2_m` AHB master interface, are on separate clocks that must be driven by the system.

All DAP-Lite2 clocks may be asynchronous to each other.

4 SoC-600 components functional description

This chapter describes the functionality of the the SoC-600 components that are integrated into the DAP-Lite2 modules. Note that this chapter is for reference. Your configuration is determined by the DAP-Lite2 module that you use.

4.1 Debug port

The `css600_dp` module implements the JTAG and Serial Wire Debug Port protocols. Either of these protocols can be omitted to save area in systems that do not require both protocols.

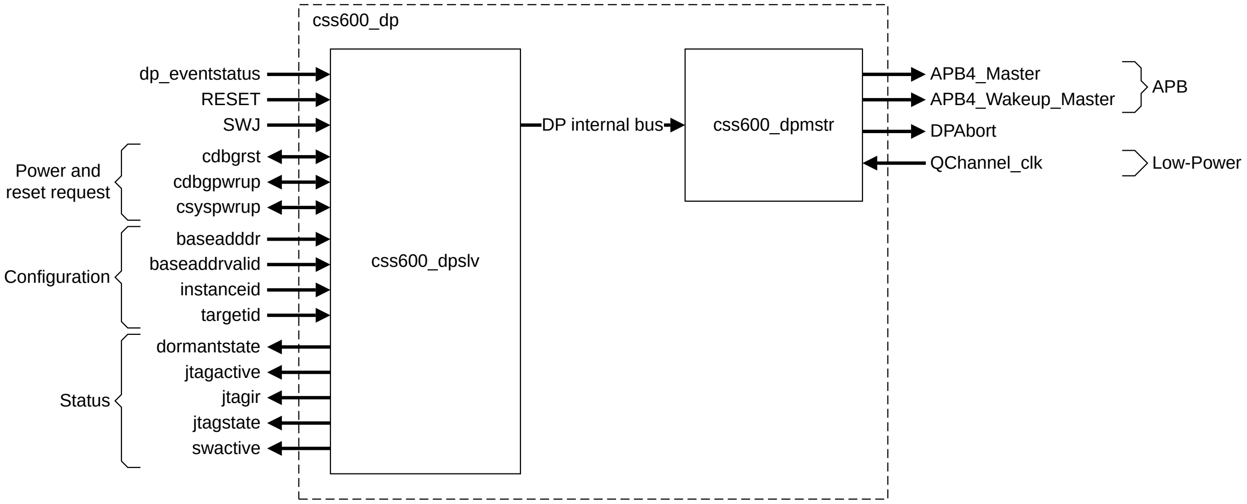
The debug port communicates with the debug components through the APB infrastructure that is connected to the debug port APB master interface.

The debug port implements the following features:

- ADIv6 architecture
- Single clock domain in each part
- Asynchronous bridge between the slave and master parts
- 4-bit or 8-bit Instruction register for JTAG implementation
- Separate slave and master components, implementing JTAG, Serial Wire, or both in the slave, and APB in the master

The following figure shows the external connections on the *Debug Port* (DP).

Figure 4-1: `css600_dp` logical connections



4.2 Memory access ports

Memory Access Ports (MEM-APs) connect one memory system to another using one of the AMBA bus protocols: AHB or APB.

The *Arm® Debug Interface Architecture Specification ADIv6.0* defines a MEM-AP. This definition provides two logical views of the access port to the debugger. These two views are referred to as twin APs or logical APs. In SoC-600, these two logical APs are contiguous in the memory map and each one of them occupies 4KB address space. An external debugger can only discover one of the twin APs through the ROM table. The other AP is dedicated for self-hosted debug. The MEM-AP itself is not capable of differentiating which of the twin APs is visible in the ROM table. The MEM-AP decodes the access requests on the APB slave interface and maps them to AP-L0 or AP-L1, based on the value of **paddr_s[12]**.

Note

`daplite2_m` and `daplite2_ar` do not have twin APs.

4.2.1 APB Access Port

The `css600_apbap` module is a *Memory Access Port* (MEM-AP). The `css600_apbap` is an APB4 slave component that provides access to another APB4 memory system.

Use the `css600_apbap` to provide access to an APB4 memory space, for example:

- A subsystem of CoreSight™ components that includes Arm® Cortex®-A or Cortex®-R processors
- A subsystem of CoreSight™ components
- Any other APB4 memory system

The APB Access Port allows visibility into another memory system from the debug APB infrastructure.

The APB-AP provides an AMBA APB4 slave interface for programming and an AMBA APB4 master interface for accessing the target memory system. The programmers model contains the details of the registers for accessing the features of the APB4 master interface.

The APB-AP provides the following features:

- Error response
- Stalling accesses
- Little-endian only
- Single clock domain

- 32-bit data access only
- Auto-incrementing *Transfer Address Register* (TAR)
- An APB4 slave interface
- An APB4 master interface
- An Access Port Enable interface
- CoreSight Component base pointer register
- A Q-Channel LPI for high-level clock management

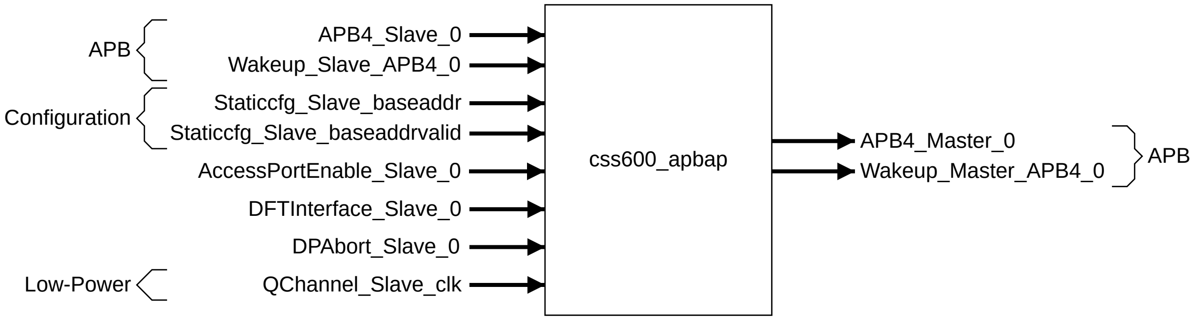
The APB-AP does not support subword write transfers.

Note

If the DP issues an abort over the Debug APB interface, the APB-AP completes the transaction on its Debug APB slave interface immediately. The DAP transfer abort does not cancel the ongoing APB transfer on the APB master interface.

The following figure shows the external connections on the APB Access Port.

Figure 4-2: `css600_apbap` logical connections



4.2.2 AHB Access Port

The `css600_ahbap` module is a *Memory Access Port* (MEM-AP). The `css600_ahbap` is an APB4 slave component that provides access to an AHB5 memory system.

Use the `css600_ahbap` to provide access to an AHB5 memory space, for example:

- An Arm® Cortex®-M processor and subsystem
- Any other AHB5 memory system

The AHB Access Port allows visibility into another memory system from the debug APB infrastructure.

The AHB-AP provides an AMBA® APB4 slave interface for programming and an AMBA® AHB5 master interface for accessing the target memory system. The programmers model contains the details of the registers for accessing the features of the AHB master interface.

The AHB-AP provides the following features:

- Error response
- Stalling accesses
- Little-endian only
- Single clock domain
- Auto-incrementing *Transfer Address Register* (TAR)
- An APB4 slave interface
- An AHB5 master interface
- An Access Port Enable interface
- 8-bit, 16-bit, or 32-bit data access
- CoreSight Component base pointer register
- Support for AHB5 TrustZone® signaling
- A Q-Channel LPI for high-level clock management

The AHB-AP does not support:

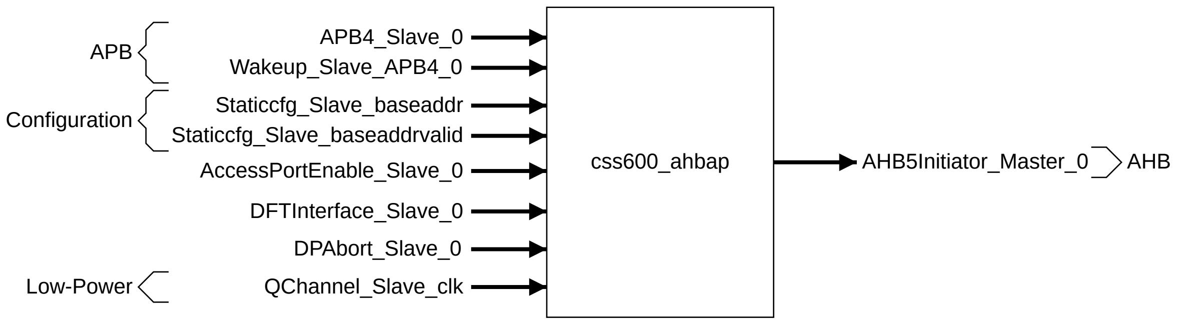
- Exclusive accesses
- Unaligned transfers
- BURST or SEQ transactions

Note

If the DP issues an abort over the Debug APB interface, the AHB-AP completes the transaction on its Debug APB slave interface immediately. The DAP transfer abort does not cancel the ongoing AHB transfer.

The following figure shows the external connections on the AHB Access Port.

Figure 4-3: `css600_ahbap` logical connections



4.2.3 Error response handing

CoreSight™ DAP-Lite2 *Memory Access Ports* (MEM-APs) implement Error Response Handling Version 1.

Error Response Handling V1 is defined in the *Arm® Debug Interface Architecture Specification ADIv6.0*. Support for this error handling mechanism is indicated in the CFG.ERR register field. The three register bits CSW.ERRNPASS, CSW.ERRSTOP, and TRR.ERR are used to define the behavior of this feature. See the relevant programmers model register descriptions for more information.

The MEM-AP logs errors in Transfer Response Register by setting TRR.ERR bit to 1. When set, this bit remains set until software clears it by writing 1 to it. The following types of memory access errors are logged:

Authentication failure
This error is due to an unauthenticated memory access attempt, such as:

- Any memory access when **ap_en** is LOW
- A Secure memory access when **ap_secure_en** is LOW

Stopped on error
This error is due to a memory access attempt when TRR.ERR=1 and CSW.ERRSTOP=1.

AHB/APB error
This is an error response that is received on the AP master interface indicating that the memory access failed.

Abort
These are aborted memory transfers.

Master busy
This error happens if a memory access is attempted after an abort, but while the CSW.TrInProg bit is still set.

Internal register access errors are not logged in the TRR but are always passed on the APB slave interface. If a register write is attempted after an abort while the CSW.TrInProg bit is set, an error is generated.

The register bit CSW.ERRNPASS controls whether a memory access error is passed back to the requestor. The internal register access errors are always passed back on the APB slave interface regardless of the value of this bit.

The CSW.ERRNPASS bit has the following effect on behavior:

0
Memory access errors are passed back on the APB slave interface.

1
Memory access errors are not passed back on the APB slave interface. In this case, a normal APB response is returned even for failed memory transactions.

There are exceptions to this rule. In these cases, the error is always passed on the APB slave interface, regardless of the status of the CSW.ERRNPASS bit. The exceptions are:

- If the memory transaction is aborted
- If the error is generated due to a memory access attempt, while the CSW.TrInProg bit is still set from a previously aborted access

The APB read data for all transactions that generate an error is **UNKNOWN**.

If no previous memory access errors are logged, that is TRR.ERR=0, memory accesses are allowed, regardless of the state of CSW.ERRSTOP.

If a previous memory access error is still logged, that is TRR.ERR=1, the register field CSW.ERRSTOP controls whether to prevent memory accesses as follows:

- If CSW.ERRSTOP is programmed as 0, new memory accesses are allowed.
- If CSW.ERRSTOP is programmed as 1, no new memory accesses are allowed. Any new memory accesses result in an error response on the APB slave interface, provided CSW.ERRNPASS is 0. In this case, TRR.ERR remains set and the memory transfer is not initiated.

The following table summarizes this MEM-AP behavior for memory errors other than Abort and Master Busy.

Table 4-1: MEM-AP behavior for memory errors other than Abort and Master Busy

TRR.ERR	CSW.ERRNPASS	CSW.ERRSTOP	New memory access	Slave error	Error logged
0	0	x	Allowed if Authenticated by the Access Port Enable interface, otherwise blocked.	Passed	Yes
0	1	x		Not passed	Yes
1	0	0		Passed	Yes
1	1	0		Not passed	Yes
1	0	1	Blocked	Passed	Yes
1	1	1	Blocked	Not passed	Yes

The twin logical APs implement error handling independently, and the errors that are received or generated on one do not affect the other.

Note

`daplite2_m` and `daplite2_ar` do not have twin APs.

Memory errors, other than Abort and Master-Busy, are maskable errors. That is, they can be masked from appearing on an APB slave interface by setting the CSW.ERRNPASS bit. It is possible for a single memory access to cause multiple error sources to generate errors at the same time. For example, a memory access can trigger a stop-on-error and an authentication failure.

If an error is masked, an error response is passed on the APB slave interface, even if CSW.ERRNPASS is 1, and if at least one of the sources of error is non-maskable (Abort or Master-Busy). If all the triggered error sources are maskable, the error is passed only if CSW.ERRNPASS is 0.

If the Access Port Enable interface signals change while a memory transfer is in progress, the MEM-AP still completes the ongoing transfer normally. The new Access Port Enable interface values then take effect from the next transaction. If a memory access request is received while the

MEM-AP is in Q_STOPPED state, the authentication signal values are sampled only after entering Q_RUN state in the first cycle, and that value is used to determine whether to allow or block the pending APB transfer.

4.3 APB interconnect

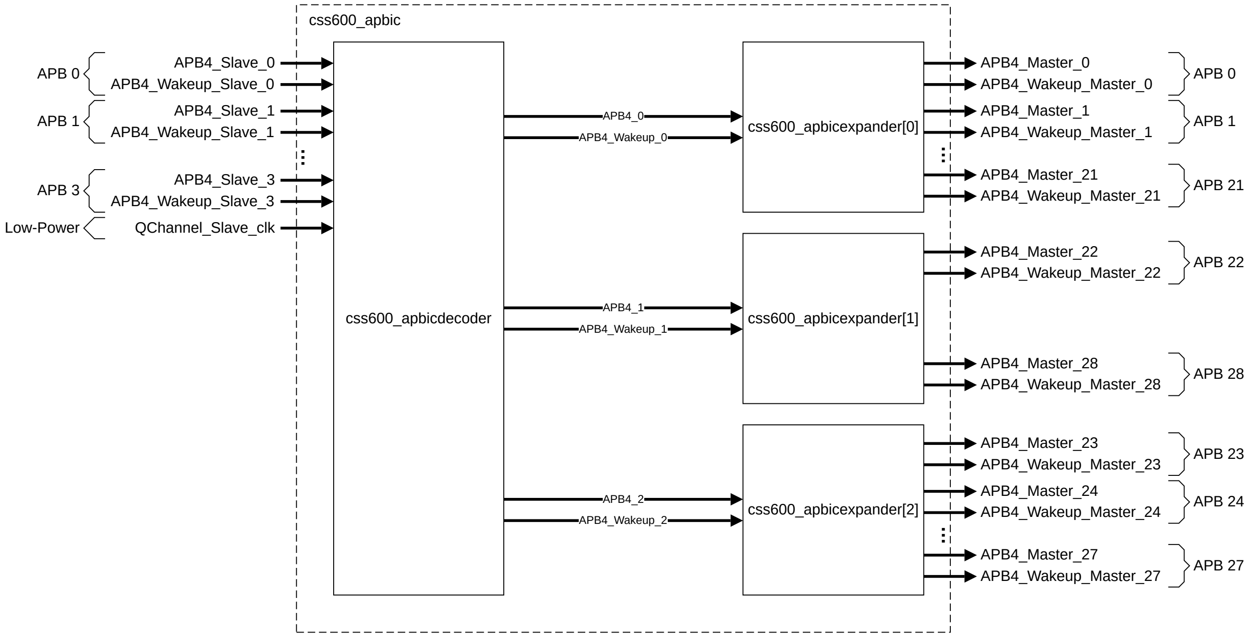
The `css600_apbic` is used to provide connections between APB4 masters and APB4 slaves anywhere in a CoreSight system.

APB4 masters can be debug ports, APB Access Ports, or other APB masters from a compute subsystem. It is a two-part meta-component that has the following features:

- Single clock domain
- Decoder component configurable for up to four slave interfaces and up to 64 master interfaces
- Physical grouping of decoded master interfaces using one or more configurable expander components
- Option to insert APB asynchronous or synchronous bridges between decoder and expander instances to cross power and clock domain boundaries
- Configurable APB address widths to suit addressable ranges
- A Q-Channel LPI for high-level clock management

The following figure shows the external connections on the APB interconnect.

Figure 4-4: `css600_apbic` logical connections



4.3.1 Arbitration

The internal arbiter arbitrates between competing slave interfaces for access to the debug APB.

When a slave interface raises a request, the arbiter gives the highest priority to the slave interface with the lowest instance suffix. For example, Slave Interface 0 > Slave Interface 1 > Slave Interface 2 > Slave Interface 3. The order in which the slave interfaces raised their requests relative to each other is not used in arbitration.

The arbitration is re-evaluated after every access.

4.3.2 Error response

The APB interconnect returns an error on its slave interface under certain conditions.

An error response is returned when either:

- The targeted APB slave returns an error response
- A slave interface accesses an address that does not decode to any connected APB slave

4.4 APB ROM Table

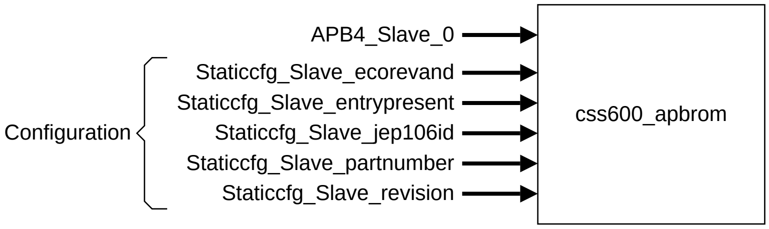
The `css600_apbrom` module is a *ROM Table* with an APB4 slave interface.

Use the `css600_apbrom` to:

- Identify part of your system or subsystem.
- Indicate the locations of other CoreSight™ components in the same address space to an External Debugger.

The following figure shows the external connections on the APB ROM Table.

Figure 4-5: `css600_apbrom` logical connections



4.5 APB asynchronous bridge

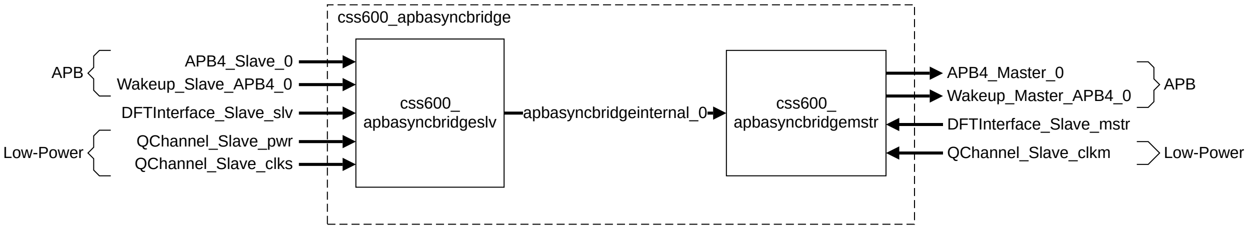
The `css600_apbasyncbridge` is used where an AMBA APB4 bus is required to cross a clock or power domain boundary.

The APB asynchronous bridge provides the following features:

- Two independent clock domains with any phase or frequency alignment
- Two independent power domains, either of which can be switched relative to the other
- Three Q-Channel LPIs for slave side clock, master side clock, and power switching management
- A two-part meta-component with separate slave and master side components
- Configurable APB address width
- Configurable 2- or 3-deep synchronizers

The following figure shows the external connections on the APB asynchronous bridge.

Figure 4-6: `css600_apbasyncbridge` logical connections



4.6 APB3 to APB4 adapter

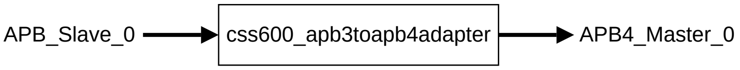
The `css600_apb3toapb4adapter` is an IP-XACT phantom component that is provided to support stitching in an IP-XACT tooling product.

There is no Verilog module for `css600_apb3toapb4adapter`.

Use the `css600_apb3toapb4adapter` to connect an APB3 master to an APB4 slave interface.

The following figure shows the external connections on the APB3 to APB4 adapter.

Figure 4-7: `css600_apb3toapb4adapter` logical connections



4.7 APB4 to APB3 adapter

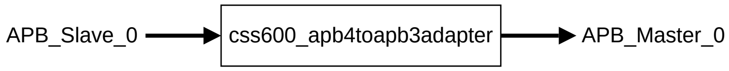
The `css600_apb4toapb3adapter` is an IP-XACT phantom component that is provided to support stitching in an IP-XACT tooling product.

There is no Verilog module for `css600_apb4toapb3adapter`.

Use the `css600_apb4toapb3adapter` to connect an APB4 master to an APB3 slave interface.

The following figure shows the external connections on the APB4 to APB3 adapter.

Figure 4-8: `css600_apb4toapb3adapter` logical connections



5 DAP-Lite2 programmers model

This chapter describes the values and configuration details that are specific to the CoreSight™ DAP-Lite2 modules.

5.1 DAP-Lite2 for Arm Cortex-A and Cortex-R processors

This section describes the DAP-Lite2 AR programmers model.

5.1.1 Debug Port for Cortex-A and Cortex-R

The following tables list values and configuration details that are specific to the `daplite2_ar`.

Register.Field	Value	Description
DPIDR1.ASIZE	`0x0C`	12-bit address

Note

The DP is configured with a 12-bit address space, so only the first 4KB of the APB-AP registers are accessible. `daplite2_ar` does not support multiple masters accessing the APB-AP.

Register.Field	Value	Description
BASEPTR0.PTR	`0x000`	The DP points to the APB-AP
BASEPTR0.VALID	`0b1`	Base Address is valid

Register.Field	Value	Description
TARGETID.TREVISION	IMPLEMENTATION DEFINED	Driven from *`dp_targetid`*
TARGETID.TPARTNO	IMPLEMENTATION DEFINED	Driven from *`dp_targetid`*
TARGETID.TDESIGNER	IMPLEMENTATION DEFINED	Driven from *`dp_targetid`*
DLPIDR.TINSTANCE	IMPLEMENTATION DEFINED	Driven from *`dp_instanceid`*

These values depend on the integration of `daplite2_ar` in the system. They should identify the SoC and its designer. If you require the values of these fields, consult the system integrator.

5.1.2 APB Access Port

The following table lists values and configuration details that are specific to the `daplite2_ar`.

Register.Field	Value	Description
BASE.BASEADDR	IMPLEMENTATION DEFINED	Defined by configuration parameter *`ROM_BASE_ADDR`*
BASE.EntryPresent	`0b1`	ROM Entry points to a CoreSight component

The APB Access Port is configured to point to the ROM Table inside `daplite2_ar`. If you require the values of these fields, consult the system integrator.

5.1.3 APB ROM table

The following table lists values and configuration details that are specific to the `daplite2_ar`.

Register.Field	Value	Description
DEVID.SYSMEM	`0b0`	Dedicated debug bus
PIDR4.DES_2	IMPLEMENTATION DEFINED	Driven from *`rom_jep106_id`*
PIDR2.REVISION	IMPLEMENTATION DEFINED	Driven from *`rom_revision`*
PIDR2.DES_1	IMPLEMENTATION DEFINED	Driven from *`rom_jep106_id`*
PIDR1.DES_0	IMPLEMENTATION DEFINED	Driven from *`rom_jep106_id`*
PIDR1.PART_1	IMPLEMENTATION DEFINED	Driven from *`rom_part_number`*
PIDR0.PART_0	IMPLEMENTATION DEFINED	Driven from *`rom_part_number`*

These values depend on the integration of `daplite2_ar` in the system. They should identify the SoC and its designer. If you require the values of these fields, consult the system integrator.

5.1.4 Memory map

The following figures show the `daplite2_ar` block diagram and memory maps.

Figure 5-1: daplite2_ar block diagram

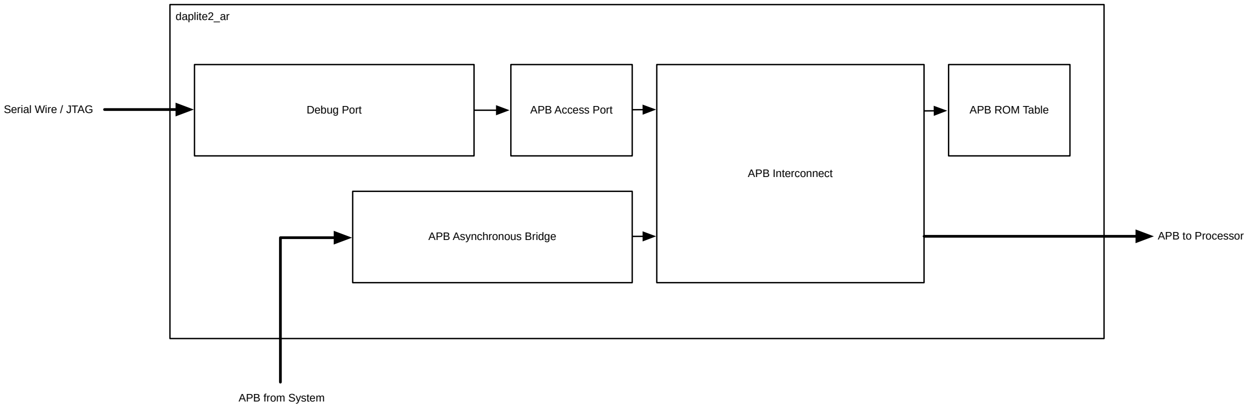


Figure 5-2: daplite2_ar DP memory map

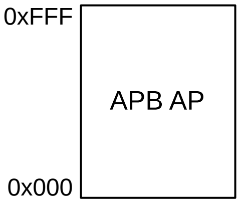
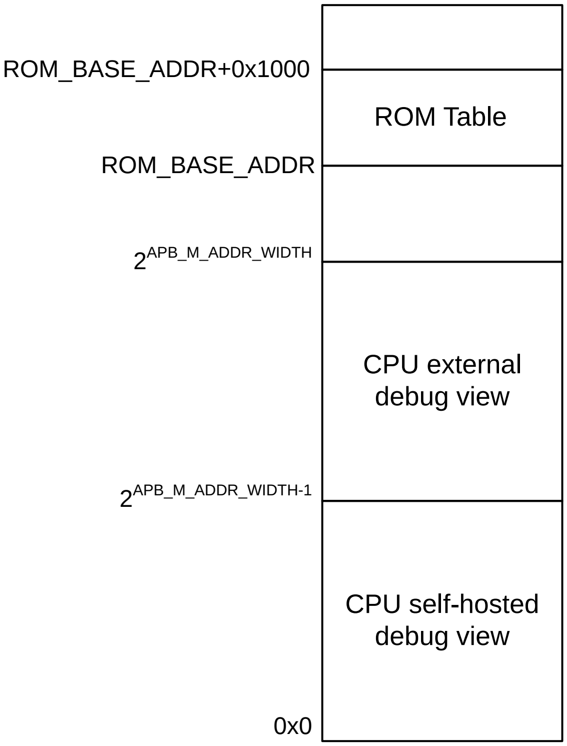


Figure 5-3: daplite2_ar AP memory map



5.2 DAP-Lite2 for Arm Cortex-M processors

This section describes the DAP-Lite2 M programmers model.

5.2.1 Debug Port for Cortex-M

The following table lists values and configuration details that are specific to the `daplite2_m`.

Register.Field	Value	Description
DPIDR1.ASIZE	`0x0C`	12-bit address

Note

The DP is configured with a 12-bit address space, so only the first 4KB of the AHB-AP registers are accessible. `daplite2_m` does not support multiple masters accessing the AHB-AP.

Register.Field	Value	Description
BASEPTR0.PTR	`0x000`	DP points to the AHB-AP
BASEPTR0.VALID	`0b1`	Base Address is valid

Register.Field	Value	Description
TARGETID.TREVISION	IMPLEMENTATION DEFINED	Driven from *`dp_targetid`*
TARGETID.TPARTNO	IMPLEMENTATION DEFINED	Driven from *`dp_targetid`*
TARGETID.TDESIGNER	IMPLEMENTATION DEFINED	Driven from *`dp_targetid`*
DLPIDR.TINSTANCE	IMPLEMENTATION DEFINED	Driven from *`dp_instanceid`*

These values depend on the integration of `daplite2_m` in the system. They should identify the SoC and its designer. If you require the values of these fields, consult the system integrator.

5.2.2 AHB Access Port

The following table lists values and configuration details that are specific to the `daplite2_m`.

Register.Field	Value	Description
BASE.BASEADDR	IMPLEMENTATION DEFINED	Driven from *`ap_baseaddr`*
BASE.EntryPresent	IMPLEMENTATION DEFINED	Driven from *`ap_baseaddr_valid`*

These values depend on the integration of `daplite2_m` in the system. They should point to another CoreSight component, typically a ROM Table. If you require the values of these fields, consult the system integrator.

5.2.3 Memory map

The following figures show the `daplite2_m` block diagram and memory maps.

Figure 5-4: daplite2_m block diagram

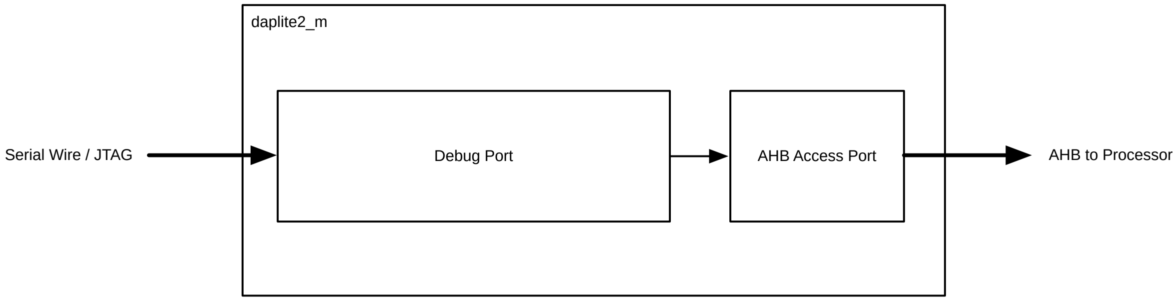


Figure 5-5: daplite2_m DP memory map

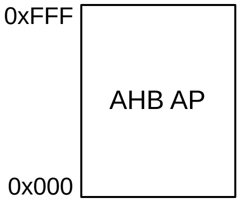
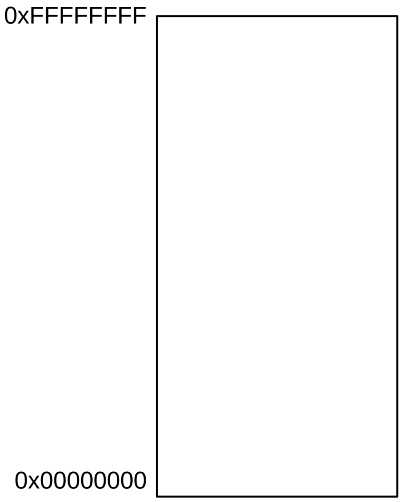


Figure 5-6: daplite2_m AP memory map



Your system integration determines the AP memory map.

6 SoC-600 components programmers model

This chapter describes the programmers models for the SoC-600 components used in DAP-Lite2 modules.

6.1 SoC-600 components programmers model

These SoC-600 programmers' models are for the components that make up the DAP-Lite2 modules.

Where the SoC-600 components have configuration dependencies or dependencies on tie-offs, you must also refer to the relevant DAP-Lite2 module information in 5 DAP-Lite2 programmers model on page 30 for details of that configuration.

The following information applies to the SoC-600 components registers:

- The base address of any component is not fixed, and can be different for any particular system implementation. The offset of each register within a component from the component base address is fixed.
- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in **UNPREDICTABLE** behavior.
- Unless otherwise stated in the accompanying text:
    - Do not modify undefined register bits.
    - Ignore undefined register bits on reads.
    - All register bits are reset to the reset value specified in the register summary table for the component.
- Access types are described as follows:

RW	Read and write
RO	Read only
WO	Write only

6.2 css600_dp introduction

This section describes the programmers model of the `css600_dp`.

The register block in the SoC-600 DP is shared between two different protocol engines, the JTAG-DP and the SW-DP.

The DP programmers model consists of the following registers. The programmers model is based on the *Arm® Debug Interface Architecture Specification ADIv6.0*. Because the DP only supports 32-

bit addressing, any read to BASEPTR1 always returns 0 and any writes to SELECT1 register are ignored.

6.2.1 `css600_dp` register summary

The following table shows the registers in offset order from the base memory address.

More than one register can appear at a given address, depending on the value of SELECT.DPBANKSEL. The combinations of address offset and SELECT.DPBANKSEL value, and whether the register is accessible by the JTAG-DP, SW-DP, or both, are all shown in the following table.

Note

A reset value containing one or more '-' means that this register contains **UNKNOWN** or **IMPLEMENTATION-DEFINED** values. See the relevant register description for more information.

A DPBANKSEL value containing an 'X' means that the DPBANKSEL value is ignored.

Locations that are not listed in the table are Reserved.

Table 6-1: `css600_dp` register summary

Offset	DPBANKSEL	Name	JTAG-DP	SW-DP	Reset	Description
-	X	IDCODE	No	No	`0x4BA06477` or `0x4BA07477`	6.2.2.1 css600_dp JTAG TAP ID Register, IDCODE on page 37
-	X	ABORT	No	Yes	`0x00000000`	6.2.2.2 css600_dp AP Abort Register, ABORT on page 38
`0x000000--`	`0x0`	DPIDR	Yes	Yes	`0x4C013477`	6.2.2.3 css600_dp Debug Port Identification Register, DPIDR on page 39
`0x0000`	`0x1`	DPIDR1	Yes	Yes	`0x000000--`	6.2.2.4 css600_dp Debug Port Identification Register 1, DPIDR1 on page 39
`0x00000000`	`0x2`	BASEPTR0	Yes	Yes	`0x-----00-`	6.2.2.5 css600_dp Base Pointer Register 0, BASEPTR0 on page 40
`0x0000`	`0x3`	BASEPTR1	Yes	Yes	`0x00000000`	6.2.2.6 css600_dp Base Pointer Register 1, BASEPTR1 on page 41
`0x0004`	`0x0`	CTRLSTAT	Yes	Yes	`0x--000000`	6.2.2.7 css600_dp Control/Status Register, CTRLSTAT on page 42
`0x0004`	`0x1`	DLCR	No	Yes	`0x00000040`	6.2.2.8 css600_dp Data Link Control Register, DLCR on page 43
`0x0004`	`0x2`	TARGETID	Yes	Yes	`0x--------`	6.2.2.9 css600_dp Target Identification Register, TARGETID on page 44
`0x0004`	`0x3`	DLPIDR	Yes	Yes	`0x-0000001`	6.2.2.10 css600_dp Data Link Protocol Identification Register, DLPIDR on page 45
`0x0004`	`0x4`	EVENTSTAT	Yes	Yes	`0x0000000-`	6.2.2.11 css600_dp Event Status Register, EVENTSTAT on page 45

Offset	DPBANKSEL	Name	JTAG-DP	SW-DP	Reset	Description
0x0004	0x5	SELECT1	Yes	Yes	0x00000000	6.2.2.12 css600_dp Select Register 1, SELECT1 on page 46
0x0008	X on reads	RESEND	No	Yes	0x00000000	6.2.2.13 css600_dp Read Resend Register, RESEND on page 47
0x0008	X on writes	SELECT	Yes	Yes	0x00000000	6.2.2.14 css600_dp Select Register, SELECT on page 48
0x000C	X on reads	RDBUFF	No	Yes	0x00000000	6.2.2.15 css600_dp Read Buffer Register, RDBUFF on page 48
0x000C	X on writes	TARGETSEL	No	Yes	0x00000000	6.2.2.16 css600_dp Target Selection Register, TARGETSEL on page 49

6.2.2 css600_dp register descriptions

This section describes the css600_dp registers.

6.2.1 css600_dp register summary on page 36 provides cross references to individual registers.

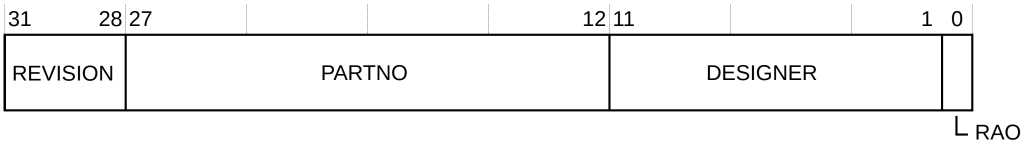
6.2.2.1 css600_dp JTAG TAP ID Register, IDCODE

The IDCODE value enables a debugger to identify the JTAG DP to which it is connected.

JTAG-DP Access is through its own scan-chain using the IDCODE instruction in the JTAG IR. SW-DP There is no IDCODE register in the SW-DP.

The following figure shows the bit assignments.

Figure 6-1: IDCODE register bit assignments



The following table shows the bit assignments.

Table 6-2: IDCODE register bit assignments

Bits	Reset value	Name	Function
[31:28]	0x4	REVISION	Revision. An incremental value starting at 0x0 for the first design of a component. See the Component list in Chapter 1 for information on the RTL revision of the component.

Bits	Reset value	Name	Function
[27:20]	**IMPLEMENTATION DEFINED**	PARTNO	Part Number of the DP. The value depends on the Instruction Register length configuration of the `css600_dp`: **`0xBA06`** 4-bit IR **`0xBA07`** 8-bit IR
[11:1]	`0x23B`	DESIGNER	Designer ID based on 11-bit JEDEC JEP106 continuation and identity code `0x23B`, Arm Ltd
[0]	`0b1`	RAO	RAO

See *Arm® Debug Interface Architecture Specification ADIv6.0* for details about accessing the IDCODE value in a JTAG DP.

6.2.2.2 `css600_dp` AP Abort Register, ABORT

The ABORT register drives the **dp_abort** pin on the DP, which goes to APs to abort the current transaction.

JTAG-DP Access is through its own scan-chain using the ABORT instruction in the JTAG IR.

SW-DP Access is by a write to offset `0x0` of the DP register map.

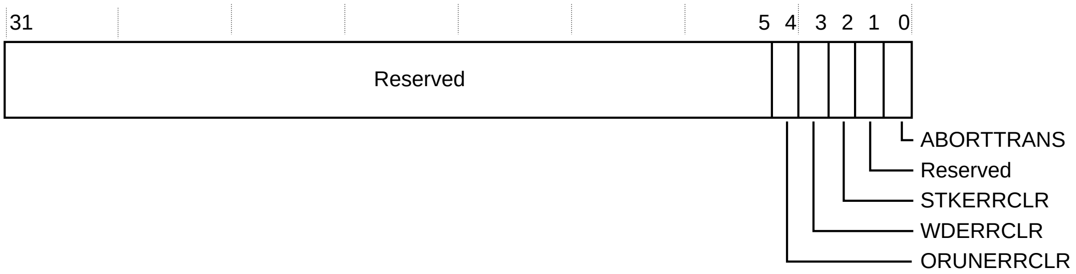
The ABORT register characteristics are:

Attributes

Offset	`0x0000`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-2: ABORT register bit assignments



The following table shows the bit assignments.

Table 6-3: ABORT register bit assignments

Bits	Reset value	Name	Function
[4]	`0b0`	ORUNERRCLR	Write 1 to this bit to clear the CTRLSTAT.STICKYORUN overrun error bit to 0
[3]	`0b0`	WDERRCLR	Write 1 to this bit to clear the CTRLSTAT.WDATAERR write data error bit to 0
[2]	`0b0`	STKERRCLR	Write 1 to this bit to clear the CTRLSTAT.STICKYERR sticky error bit to 0
[0]	`0b0`	ABORTTRANS	Write 1 to this bit to generate an abort. This aborts the current AP transaction

6.2.2.3 `css600_dp` Debug Port Identification Register, DPIDR

The DPIDR provides information about the DP.

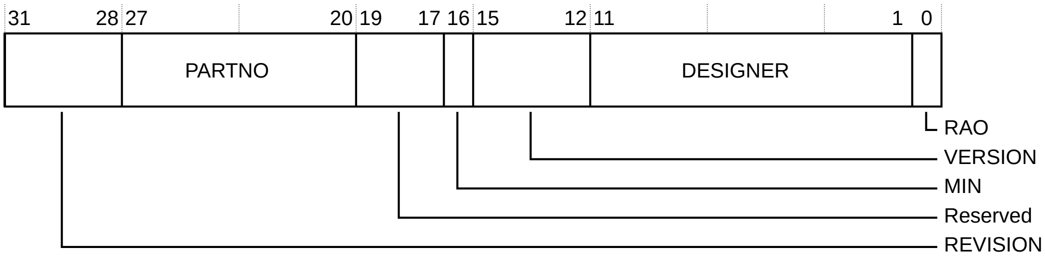
The DPIDR register characteristics are:

Attributes

Offset	`0x0000`
Type	Read-only
Reset	`0x4C013477`
Width	32

The following figure shows the bit assignments.

Figure 6-3: DPIDR register bit assignments



The following table shows the bit assignments.

Table 6-4: DPIDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0100`	REVISION	Revision code: `0b0100` - r0p4
[27:20]	`0b11000000`	PARTNO	Part Number of the DP
[16]	`0b1`	MIN	Transaction counter, Pushed-verify, and Pushed-find operations are not implemented
[15:12]	`0b0011`	VERSION	Version of DP architecture implemented: DAP-Lite2 is DPv3, so the value of this field is `0x3`
[11:1]	`0b01000111011`	DESIGNER	Designer ID based on 11-bit JEDEC JEP106 continuation and identity code: the Arm value is `0x23B` for this field
[0]	`0b1`	RAO	RAO

6.2.2.4 `css600_dp` Debug Port Identification Register 1, DPIDR1

The DPIDR1 register is the extension of DPIDR and provides information about the DP.

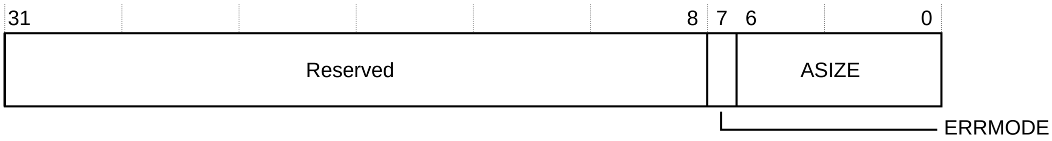
The DPIDR1 register characteristics are:

Attributes

Offset	`0x0000`
Type	Read-only
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

Figure 6-4: DPIDR1 register bit assignments



The following table shows the bit assignments.

Table 6-5: DPIDR1 register bit assignments

Bits	Reset value	Name	Function
[7]	`0b1`	ERRMODE	Error reporting mode support: **1** CTRLSTAT.ERRMODE implemented.
[6:0]	**IMPLEMENTATION DEFINED**	ASIZE	Address size. This defines the size of the address in the SELECT register, and the BASEPTR0 register. Allowed values are: **`0x0C`** 12-bit address **`0x14`** 20-bit address **`0x20`** 32-bit address All other values are reserved. This is an **IMPLEMENTATION-DEFINED** value that depends on the configuration of the component.

6.2.2.5 `css600_dp` Base Pointer Register 0, BASEPTR0

BASEPTR0 and BASEPTR1 together provide an initial system address for the first component in the system. Typically, this is the address of a top-level ROM table that indicates where APv2 APs are located. The size of the address is defined in DPIDR1.ASIZE, which defines the size of the whole address even though bits [11:0] are always zero, as the minimum address space for each component is 4KB.

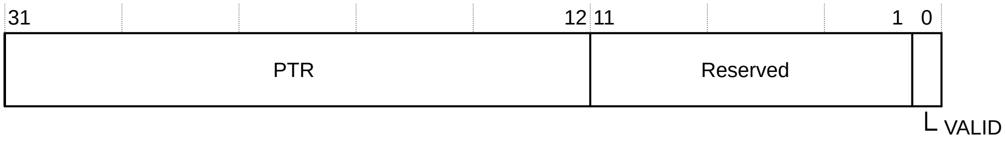
The BASEPTR0 register characteristics are:

Attributes

Offset	`0x-----00-`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

Figure 6-5: BASEPTR0 register bit assignments



The following table shows the bit assignments.

Table 6-6: BASEPTR0 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION DEFINED	PTR	Base address bits [31:12] of first component in the system. The address is aligned to a 4KB boundary. This **IMPLEMENTATION-DEFINED** value depends on the interface tie-off value of **baseaddr**.
[0]	IMPLEMENTATION DEFINED	VALID	Indicates whether the base address is valid. Depends on the interface tie-off value of **baseaddr_valid**. **0** No base address specified. PTR is **UNKNOWN**. **1** Base address is specified in PTR.

6.2.2.6 `css600_dp` Base Pointer Register 1, BASEPTR1

Since the SoC-600 supports 32-bit addressing only, BASEPTR1 always reads 0s.

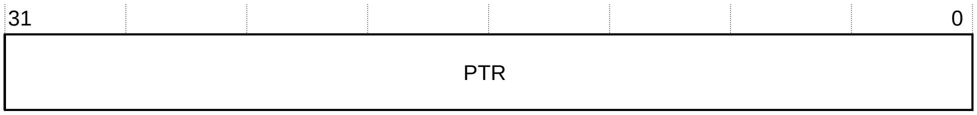
The BASEPTR1 register characteristics are:

Attributes

Offset	`0x0000`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-6: BASEPTR1 register bit assignments



The following table shows the bit assignments.

Table 6-7: BASEPTR1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	PTR	Base address bits [63:32] of first component in the system. Always reads 0s.

6.2.2.7 `css600_dp` Control/Status Register, CTRLSTAT

The Control/Status register provides control of the DP and status information about the DP.

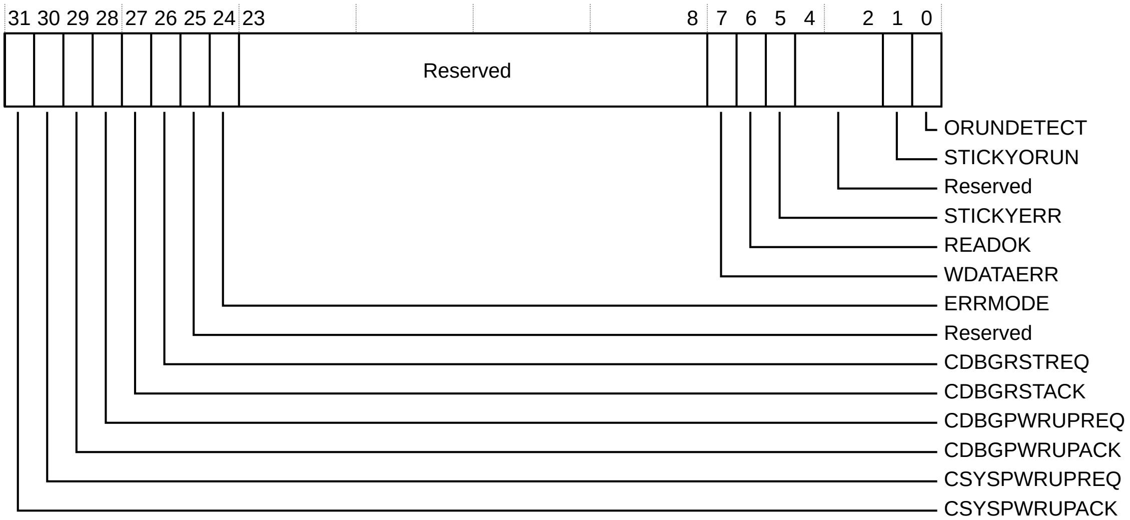
The CTRLSTAT register characteristics are:

Attributes

Offset	`0x0004`
Type	Read-write
Reset	`0x--000000`
Width	32

The following figure shows the bit assignments.

Figure 6-7: CTRLSTAT register bit assignments



The following table shows the bit assignments.

Table 6-8: CTRLSTAT register bit assignments

Bits	Reset value	Name	Function
[31]	UNKNOWN	CSYSPWRUPACK	System powerup acknowledge. Status of **CSYSPWRUPACK** interface signal.
[30]	`0b0`	CSYSPWRUPREQ	System powerup request. This bit controls the **CSYSPWRUPREQ** signal on the interface.
[29]	UNKNOWN	CDBGPWRUPACK	Debug powerup acknowledge. Status of **CDBGPWRUPACK** interface signal.
[28]	`0b0`	CDBGPWRUPREQ	Debug powerup request. This bit controls the **CDBGPRWUPREQ** signal on the interface.
[27]	UNKNOWN	CDBGRSTACK	Debug reset acknowledge. Indicates the status of the **CDBGRSTACK** signal on the interface.
[26]	`0b0`	CDBGRSTREQ	Debug reset request. This bit controls the **CDBGRSTREQ** signal on interface.
[24]	`0b0`	ERRMODE	Error Mode. **0** Errors on AP transactions set CTRLSTAT.STICKYERR **1** Errors on AP transactions do not set CTRLSTAT.STICKYERR
[7]	`0b0`	WDATAERR	This bit is DATA LINK DEFINED, such that on a JTAG-DP this bit is reserved, RES0, and on an SW-DP this bit is RO. This bit is set to 1 if a Write Data Error occurs. This happens if there is a parity or framing error on the data phase of a write, or a write that has been accepted by the DP is then discarded without being submitted to the AP. On an SW-DP, this bit is cleared to 0 by writing 1 to the ABORT.WDERRCLR bit.
[6]	`0b0`	READOK	This flag always indicates the response to the last AP read access. This bit is DATA LINK DEFINED. On JTAG-DP, the bit is reserved, RES0, and on SW-DP, access is RO. If the response to the previous AP read or RDBUFF read was OK, then the bit is set to 1. If the response was not OK, then it is cleared to 0.
[5]	`0b0`	STICKYERR	If an error is returned by an AP transaction, and CTRLSTAT.ERRMODE is b0, then this bit is set to 1. The behavior on writing is DATA LINK DEFINED: On a JTAG-DP, access is R/W1C. On a SW-DP, access is RO/WI. Clearing this bit to 0 is also DATA LINK DEFINED: On a JTAG-DP, the bit is cleared by writing 1 to this bit, or by writing 1 to the ABORT.STKERRCLR field. On SW-DP, the bit is cleared by writing 1 to the ABORT.STKERRCLR field.
[1]	`0b0`	STICKYORUN	If overrun detection is enabled, this bit is set to 1 when an overrun occurs. The behavior on writing is DATA LINK DEFINED: on a JTAG-DP, access is R/W1C. On a SW-DP, access is RO/WI. Clearing this bit to 0 is also DATA LINK DEFINED: On a JTAG-DP, the bit is cleared by writing 1 to this bit, or by writing 1 to the ABORT.ORUNERRCLR field. On SW-DP, the bit is cleared by writing 1 to the ABORT.ORUNERRCLR field.
[0]	`0b0`	ORUNDETECT	This bit is set to 1 to enable overrun detection

6.2.2.8 `css600_dp` Data Link Control Register, DLCR

The DLCR controls the operating mode of the Data link. Access to this register is **DATA LINK DEFINED**. For JTAG-DP, this register is Reserved RES0. For SW-DP, the register is as shown in the following table.

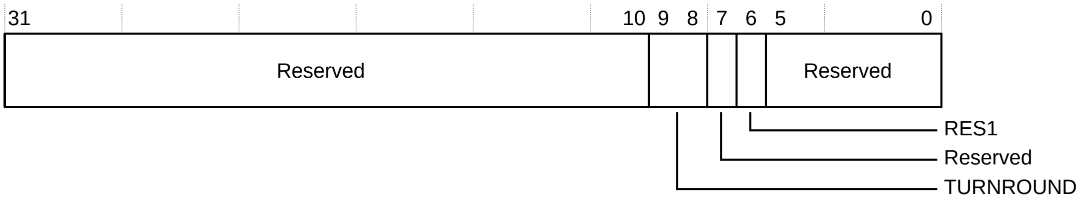
The DLCR register characteristics are:

Attributes

Offset	`0x0004`
Type	Read-write
Reset	`0x00000040`
Width	32

The following figure shows the bit assignments.

Figure 6-8: DLCR register bit assignments



The following table shows the bit assignments.

Table 6-9: DLCR register bit assignments

Bits	Reset value	Name	Function
[9:8]	0b00	TURNROUND	Turnaround tristate period: **0x0** 1 data period **0x1** 2 data periods **0x2** 3 data periods **0x3** 4 data periods
[6]	0b1	RES1	Reserved, RES1

6.2.2.9 `css600_dp` Target Identification Register, TARGETID

The TARGETID register provides information about the target when the host is connected to a single device.

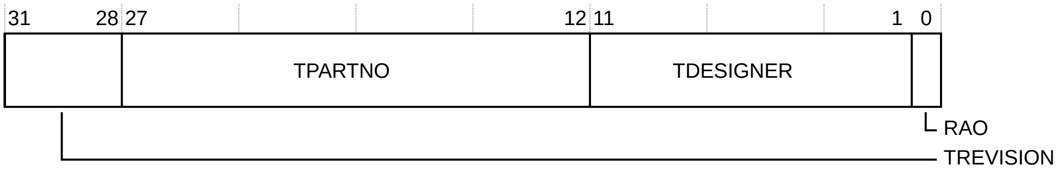
The TARGETID register characteristics are:

Attributes

Offset	0x0004
Type	Read-only
Reset	0x--------
Width	32

The following figure shows the bit assignments.

Figure 6-9: TARGETID register bit assignments



The following table shows the bit assignments.

Table 6-10: TARGETID register bit assignments

Bits	Reset value	Name	Function
[31:28]	IMPLEMENTATION DEFINED	TREVISION	Target revision. The value comes from the tie-off signal **targetid[31:28]**.
[27:12]	IMPLEMENTATION DEFINED	TPARTNO	Target part number. The value comes from the tie-off signal **targetid[27:12]**.
[11:1]	IMPLEMENTATION DEFINED	TDESIGNER	Designer ID, based on 11-bit JEDEC JEP106 continuation and identity code. The value comes from the tie-off signal **targetid[11:1]**.
[0]	`0b1`	RAO	Reserved, RAO

6.2.2.10 `css600_dp` Data Link Protocol Identification Register, DLPIDR

The DLPIDR provides protocol version information. The contents of this register are **DATA LINK DEFINED**.

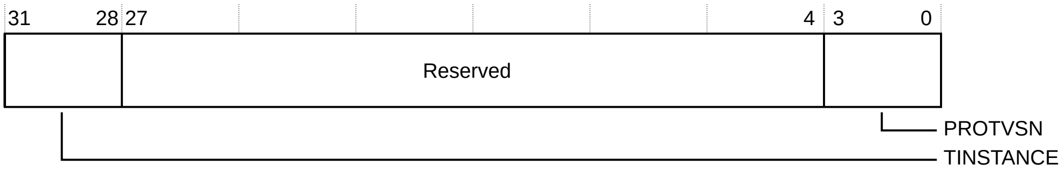
The DLPIDR register characteristics are:

Attributes

Offset	`0x0004`
Type	Read-only
Reset	`0x-00000001`
Width	32

The following figure shows the bit assignments.

Figure 6-10: DLPIDR register bit assignments



The following table shows the bit assignments.

Table 6-11: DLPIDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	IMPLEMENTATION DEFINED	TINSTANCE	The instance number for this device. For JTAG-DP: Reserved, RES0, and for SW-DP: The value comes from the tie-off signal **instanceid[3:0]**. Must be unique in a multi-drop system.
[3:0]	`0b0001`	PROTVSN	Defines the protocol version that is implemented. For JTAG-DP: `0x1`, as JTAG protocol version 1 is implemented, and for SW-DP: `0x1` as SW protocol version 2 is implemented.

6.2.2.11 `css600_dp` Event Status Register, EVENTSTAT

The EVENTSTAT register is used by the system to signal an event to the external debugger.

DAP-Lite2 implements the EVENTSTAT register with top-level input **dp_eventstatus**, connected to an output trigger of a CoreSight *Cross-Trigger Interface* (CTI) with software acknowledge. This input signal **dp_eventstatus** coming from CTI trigout is inverted and synchronized in the DP before it goes to the EVENTSTAT register.

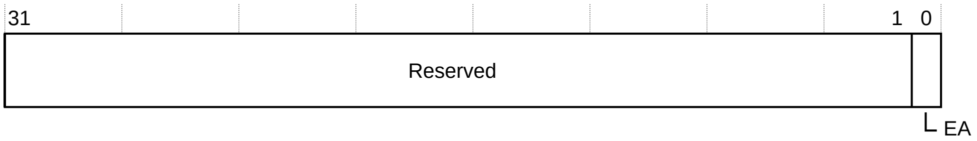
The EVENTSTAT register characteristics are:

Attributes

Offset	`0x0004`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

Figure 6-11: EVENTSTAT register bit assignments



The following table shows the bit assignments.

Table 6-12: EVENTSTAT register bit assignments

Bits	Reset value	Name	Function
[0]	UNKNOWN	EA	Event status flag. Valid values for this bit are: **0** An event requires attention **1** There is no event requiring attention

6.2.2.12 `css600_dp` Select Register 1, SELECT1

The SELECT1 register is not used as CoreSight™ DAP-Lite2 only supports 32-bit addressing.

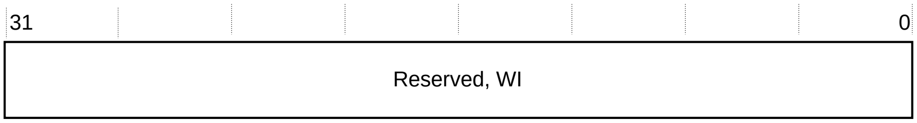
The SELECT1 register characteristics are:

Attributes

Offset	`0x0004`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-12: SELECT1 register bit assignments



The following table shows the bit assignments.

Table 6-13: SELECT1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	0x0	Reserved	Not used

6.2.2.13 css600_dp Read Resend Register, RESEND

The RESEND register enables the read data to be recovered from a corrupted debugger transfer without repeating the original AP transfer.

For JTAG-DP, this register is Reserved and any access is RES0. For SW-DP, a read to this register does not capture new data from the AP, but returns the value that was returned by the last AP read or DP RDBUFF read.

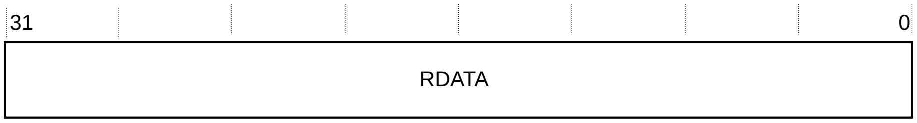
The RESEND register characteristics are:

Attributes

Offset	0x0008
Type	Read-only
Reset	0x00000000
Width	32

The following figure shows the bit assignments.

Figure 6-13: RESEND register bit assignments



The following table shows the bit assignments.

Table 6-14: RESEND register bit assignments

Bits	Reset value	Name	Function
[31:0]	0x0	RDATA	The register can only be accessed when the DP is in SW-DP configuration. Returns last AP read or DP RDBUFF read.

6.2.2.14 `css600_dp` Select Register, SELECT

The SELECT register selects the DP address bank, and also provides the address for other components in the system, which is used by the APB Master interface on the DP to drive the APB address line.

The address on the address line driven by SELECT register is set at the start of the transfer, and does not change until the next transfer. DPIDR1.ASIZE indicates the width, in bits, of the APB master interface address bus. It is defined by the configuration parameter *`APB_ADDR_WIDTH`*. The DP can only issue word-aligned addresses, so **paddr[1:0]** are always zero. Bits [3:2] come from APACC, and higher order bits come from the SELECT register. The size of the address in SELECT is defined in DPIDR1.ASIZE. Unimplemented address bits are WI.

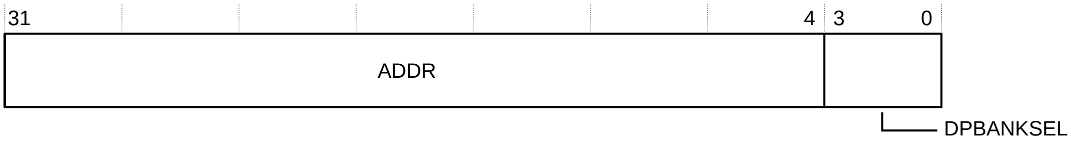
The SELECT register characteristics are:

Attributes

Offset	`0x0080`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-14: SELECT register bit assignments



The following table shows the bit assignments.

Table 6-15: SELECT register bit assignments

Bits	Reset value	Name	Function
[31:4]	`0x0`	ADDR	Address Output bits [31:4]. Selects a four-word bank of system locations to access. Address bits [3:2] are provided with APACC transactions.
[3:0]	`0b0000`	DPBANKSEL	Debug Port Address bank select

6.2.2.15 `css600_dp` Read Buffer Register, RDBUFF

The RDBUFF register captures data from the AP, presented as the result of a previous read.

Access to this register is **DATA LINK DEFINED**. On JTAG-DP, Read Buffer is always RAZ/WI. On SW-DP, the behavior is as follows.

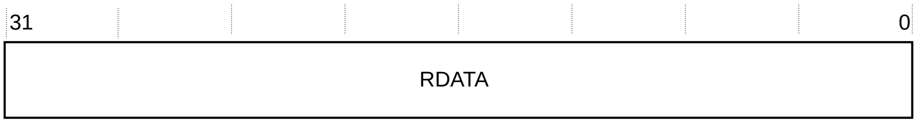
The RDBUFF register characteristics are:

Attributes

Offset	`0x000C`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-15: RDBUFF register bit assignments



The following table shows the bit assignments.

Table 6-16: RDBUFF register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	RDATA	Performing a read of the Read Buffer captures data from the AP, presented as the result of a previous read, without initiating a new AP transaction. This means that reading the Read Buffer returns the result of the last AP read access, without generating a new AP access. After you have read the DP Read Buffer, its contents are no longer valid. The result of a second read of the DP Read Buffer returns the result of the last AP read access.

6.2.2.16 `css600_dp` Target Selection Register, TARGETSEL

The TARGETSEL register selects the target device in a Serial Wire Debug multi-drop system. On a JTAG-DP, any access to this register is reserved, RES0. For SW-DP, the register is as shown in the description.

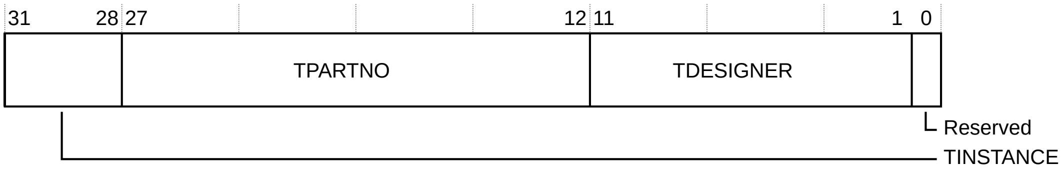
The TARGETSEL register characteristics are:

Attributes

Offset	`0x000C`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-16: TARGETSEL register bit assignments



The following table shows the bit assignments.

Table 6-17: TARGETSEL register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0000`	TINSTANCE	SW-DP: Instance number for this device. Must be unique in a multi-drop system. Must match DLPIDR.TINSTANCE.
[27:12]	`0x0`	TPARTNO	Target part number. Must match TARGETID.TPARTNO.
[11:1]	`0b00000000000`	TDESIGNER	Designer ID. Must match TARGETID.TDESIGNER.

6.3 css600_apbap introduction

This section describes the programmers model of the `css600_apbap`.

6.3.1 `css600_apbap` register summary

The following table shows the registers in offset order from the base memory address.

Note

A reset value containing one or more '-' means that this register contains **UNKNOWN** or **IMPLEMENTATION-DEFINED** values. See the relevant register description for more information.

Locations that are not listed in the table are Reserved.

The 8KB memory map contains two views of the registers, one starting at `0x00000000`, and the other at `0x00001000`.

Only 4KB can be accessed as there is no twin AP.

In the case of RW registers, the two views provide independent physical registers. Writing to a RW register in one view does not affect the contents of the same register in the other view. For all read-only registers, the two views provide read access to the same physical register. In this case, reading from either view results in the same data being read.

Table 6-18: `css600_apbap - APB4_Slave_0` register summary

Offset	Name	Type	Reset	Width	Description
0x0000	DAR0	RW	0x--------	32	6.3.2.1 Direct Access Register 0, DAR0 on page 52
0x0004	DAR1	RW	0x--------	32	6.3.2.2 Direct Access Register 1, DAR1 on page 53
0x0008	DAR2	RW	0x--------	32	6.3.2.3 Direct Access Register 2, DAR2 on page 54
…	…	…	…	…	…
0x03FC	DAR255	RW	0x--------	32	6.3.2.4 Direct Access Register 255, DAR255 on page 54
0x0D00	CSW	RW	0x30-000-2	32	6.3.2.5 Control Status Word register, CSW on page 55
0x0D04	TAR	RW	0x--------	32	6.3.2.6 Transfer Address Register, TAR on page 57
0x0D0C	DRW	RW	0x--------	32	6.3.2.7 Data Read/Write register, DRW on page 57
0x0D10	BD0	RW	0x--------	32	6.3.2.8 Banked Data register 0, BD0 on page 58
0x0D14	BD1	RW	0x--------	32	6.3.2.9 Banked Data register 1, BD1 on page 59
0x0D18	BD2	RW	0x--------	32	6.3.2.10 Banked Data register 2, BD2 on page 59
0x0D1C	BD3	RW	0x--------	32	6.3.2.11 Banked Data register 3, BD3 on page 60
0x0D24	TRR	RW	0x00000000	32	6.3.2.12 Transfer Response Register, TRR on page 61
0x0DF4	CFG	RO	0x000101A0	32	6.3.2.13 Configuration register, CFG on page 61
0x0DF8	BASE	RO	0x-----00-	32	6.3.2.14 Debug Base Address register, BASE on page 62
0x0DFC	IDR	RO	0x34770006	32	6.3.2.15 Identification Register, IDR on page 63
0x0EFC	ITSTATUS	RW	0x00000000	32	6.3.2.16 Integration Test Status register, ITSTATUS on page 64
0x0F00	ITCTRL	RW	0x00000000	32	6.3.2.17 Integration Mode Control Register, ITCTRL on page 64
0x0FA0	CLAIMSET	RW	0x00000003	32	6.3.2.18 Claim Tag Set Register, CLAIMSET on page 65
0x0FA4	CLAIMCLR	RW	0x00000000	32	6.3.2.19 Claim Tag Clear Register, CLAIMCLR on page 66
0x0FB8	AUTHSTATUS	RO	0x000000--	32	6.3.2.20 Authentication Status Register, AUTHSTATUS on page 67
0x0FBC	DEVARCH	RO	0x47700A17	32	6.3.2.21 Device Architecture Register, DEVARCH on page 68
0x0FCC	DEVTYPE	RO	0x00000000	32	6.3.2.22 Device Type Identifier Register, DEVTYPE on page 68
0x0FD0	PIDR4	RO	0x00000004	32	6.3.2.23 Peripheral Identification Register 4, PIDR4 on page 69
0x0FD4	PIDR5	RO	0x00000000	32	6.3.2.24 Peripheral Identification Register 5, PIDR5 on page 70
0x0FD8	PIDR6	RO	0x00000000	32	6.3.2.25 Peripheral Identification Register 6, PIDR6 on page 70
0x0FDC	PIDR7	RO	0x00000000	32	6.3.2.26 Peripheral Identification Register 7, PIDR7 on page 71
0x0FE0	PIDR0	RO	0x000000E2	32	6.3.2.27 Peripheral Identification Register 0, PIDR0 on page 72
0x0FE4	PIDR1	RO	0x000000B9	32	6.3.2.28 Peripheral Identification Register 1, PIDR1 on page 72
0x0FE8	PIDR2	RO	0x0000003B	32	6.3.2.29 Peripheral Identification Register 2, PIDR2 on page 73
0x0FEC	PIDR3	RO	0x00000000	32	6.3.2.30 Peripheral Identification Register 3, PIDR3 on page 74
0x0FF0	CIDR0	RO	0x0000000D	32	6.3.2.31 Component Identification Register 0, CIDR0 on page 74
0x0FF4	CIDR1	RO	0x00000090	32	6.3.2.32 Component Identification Register 1, CIDR1 on page 75
0x0FF8	CIDR2	RO	0x00000005	32	6.3.2.33 Component Identification Register 2, CIDR2 on page 76
0x0FFC	CIDR3	RO	0x000000B1	32	6.3.2.34 Component Identification Register 3, CIDR3 on page 76
0x1000	DAR0	RW	0x--------	32	6.3.2.1 Direct Access Register 0, DAR0 on page 52
0x1004	DAR1	RW	0x--------	32	6.3.2.2 Direct Access Register 1, DAR1 on page 53
0x1008	DAR2	RW	0x--------	32	6.3.2.3 Direct Access Register 2, DAR2 on page 54
…	…	…	…	…	…

Offset	Name	Type	Reset	Width	Description
0x13FC	DAR255	RW	0x--------	32	6.3.2.4 Direct Access Register 255, DAR255 on page 54
0x1D00	CSW	RW	0x30-000-2	32	6.3.2.5 Control Status Word register, CSW on page 55
0x1D04	TAR	RW	0x--------	32	6.3.2.6 Transfer Address Register, TAR on page 57
0x1D0C	DRW	RW	0x--------	32	6.3.2.7 Data Read/Write register, DRW on page 57
0x1D10	BD0	RW	0x--------	32	6.3.2.8 Banked Data register 0, BD0 on page 58
0x1D14	BD1	RW	0x--------	32	6.3.2.9 Banked Data register 1, BD1 on page 59
0x1D18	BD2	RW	0x--------	32	6.3.2.10 Banked Data register 2, BD2 on page 59
0x1D1C	BD3	RW	0x--------	32	6.3.2.11 Banked Data register 3, BD3 on page 60
0x1D24	TRR	RW	0x00000000	32	6.3.2.12 Transfer Response Register, TRR on page 61
0x1DF4	CFG	RO	0x000101A0	32	6.3.2.13 Configuration register, CFG on page 61
0x1DF8	BASE	RO	0x-----00-	32	6.3.2.14 Debug Base Address register, BASE on page 62
0x1DFC	IDR	RO	0x34770006	32	6.3.2.15 Identification Register, IDR on page 63
0x1EFC	ITSTATUS	RW	0x00000000	32	6.3.2.16 Integration Test Status register, ITSTATUS on page 64
0x1F00	ITCTRL	RW	0x00000000	32	6.3.2.17 Integration Mode Control Register, ITCTRL on page 64
0x1FA0	CLAIMSET	RW	0x00000003	32	6.3.2.18 Claim Tag Set Register, CLAIMSET on page 65
0x1FA4	CLAIMCLR	RW	0x00000000	32	6.3.2.19 Claim Tag Clear Register, CLAIMCLR on page 66
0x1FB8	AUTHSTATUS	RO	0x000000--	32	6.3.2.20 Authentication Status Register, AUTHSTATUS on page 67
0x1FBC	DEVARCH	RO	0x47700A17	32	6.3.2.21 Device Architecture Register, DEVARCH on page 68
0x1FCC	DEVTYPE	RO	0x00000000	32	6.3.2.22 Device Type Identifier Register, DEVTYPE on page 68
0x1FD0	PIDR4	RO	0x00000004	32	6.3.2.23 Peripheral Identification Register 4, PIDR4 on page 69
0x1FD4	PIDR5	RO	0x00000000	32	6.3.2.24 Peripheral Identification Register 5, PIDR5 on page 70
0x1FD8	PIDR6	RO	0x00000000	32	6.3.2.25 Peripheral Identification Register 6, PIDR6 on page 70
0x1FDC	PIDR7	RO	0x00000000	32	6.3.2.26 Peripheral Identification Register 7, PIDR7 on page 71
0x1FE0	PIDR0	RO	0x000000E2	32	6.3.2.27 Peripheral Identification Register 0, PIDR0 on page 72
0x1FE4	PIDR1	RO	0x000000B9	32	6.3.2.28 Peripheral Identification Register 1, PIDR1 on page 72
0x1FE8	PIDR2	RO	0x0000003B	32	6.3.2.29 Peripheral Identification Register 2, PIDR2 on page 73
0x1FEC	PIDR3	RO	0x00000000	32	6.3.2.30 Peripheral Identification Register 3, PIDR3 on page 74
0x1FF0	CIDR0	RO	0x0000000D	32	6.3.2.31 Component Identification Register 0, CIDR0 on page 74
0x1FF4	CIDR1	RO	0x00000090	32	6.3.2.32 Component Identification Register 1, CIDR1 on page 75
0x1FF8	CIDR2	RO	0x00000005	32	6.3.2.33 Component Identification Register 2, CIDR2 on page 76
0x1FFC	CIDR3	RO	0x000000B1	32	6.3.2.34 Component Identification Register 3, CIDR3 on page 76

6.3.2 css600_apbap register descriptions

This section describes the css600_apbap registers.

6.3.1 css600_apbap register summary on page 50 provides cross references to individual registers.

6.3.2.1 Direct Access Register 0, DAR0

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

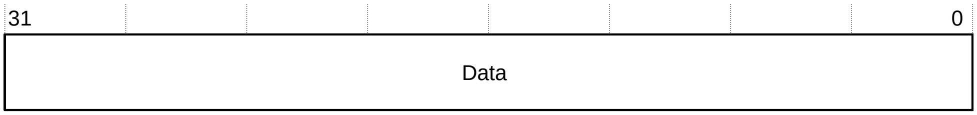
The DAR0 register characteristics are:

Attributes

Offset	`0x0000`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-17: DAR0 register bit assignments



The following table shows the bit assignments.

Table 6-19: DAR0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x0`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.3.2.2 Direct Access Register 1, DAR1

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

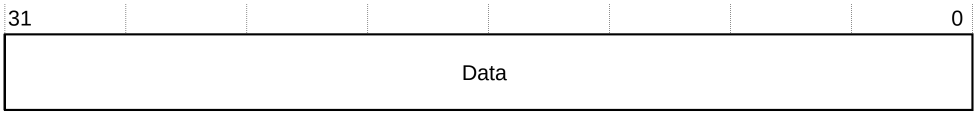
The DAR1 register characteristics are:

Attributes

Offset	`0x0004`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-18: DAR1 register bit assignments



The following table shows the bit assignments.

Table 6-20: DAR1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x4`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.3.2.3 Direct Access Register 2, DAR2

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

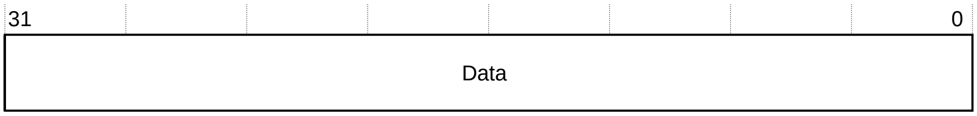
The DAR2 register characteristics are:

Attributes

Offset	`0x0008`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-19: DAR2 register bit assignments



The following table shows the bit assignments.

Table 6-21: DAR2 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x8`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.3.2.4 Direct Access Register 255, DAR255

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

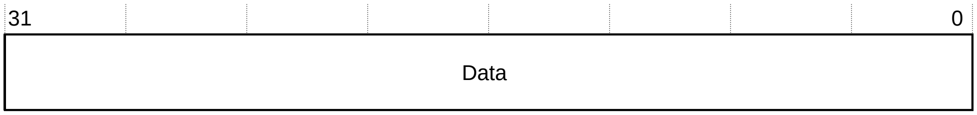
The DAR255 register characteristics are:

Attributes

Offset	`0x03FC`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-20: DAR255 register bit assignments



The following table shows the bit assignments.

Table 6-22: DAR255 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x3FC`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.3.2.5 Control Status Word register, CSW

The CSW register configures and controls accesses through the APB master interface to the connected memory system.

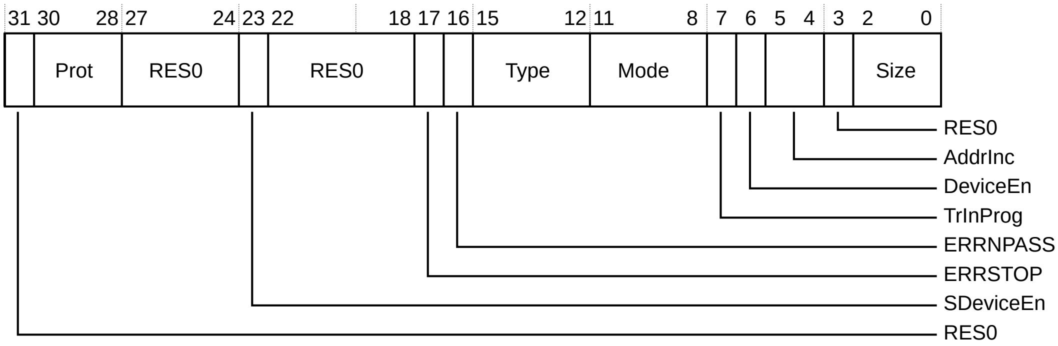
The CSW register characteristics are:

Attributes

Offset	`0x0D00`
Type	Read-write
Reset	`0x30-000-2`
Width	32

The following figure shows the bit assignments.

Figure 6-21: CSW register bit assignments



The following table shows the bit assignments.

Table 6-23: CSW register bit assignments

Bits	Reset value	Name	Function
[31]	`0b0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior.
[30:28]	`0b011`	Prot	Drives APB master interface **pprot_m[2:0]** which specifies the APB4 protection encoding. The reset value is `0x3` (Data, Non-secure, Privileged). Together with the Access Port Enable interface signals, CSW.Prot[1] determines whether a secure access is allowed on the master interface. Accesses are permitted as follows: access_permitted = (ap_en && ap_secure_en) || (ap_en && CSW.Prot[1]).
[27:24]	`0b0000`	RES0	Reserved bit or field with SBZP behavior.
[23]	UNKNOWN	SDeviceEn	Indicates the status of the **ap_en** and **ap_secure_en** ports. It is set when both **ap_en** and **ap_secure_en** are HIGH, and remains clear otherwise. If this bit is clear, Secure APB transfers are not permitted. Non-secure memory accesses and internal register accesses that do not initiate memory accesses are permitted regardless of the status of this bit.
[22:18]	`0b00000`	RES0	Reserved bit or field with SBZP behavior.
[17]	`0b0`	ERRSTOP	Stop on error. Reset to 0. **0** Memory access errors do not prevent future memory accesses **1** Memory access errors prevent future memory accesses
[16]	`0b0`	ERRNPASS	Errors are not passed upstream. **0** Memory access errors are passed upstream **1** Memory access errors are not passed upstream
[15:12]	`0b0000`	Type	This field is reserved. Reads return `0x0` and writes are ignored.
[11:8]	`0b0000`	Mode	Specifies the mode of operation. Reset to `0x0`. All other values are reserved. **`0x0`** Normal download or upload mode
[7]	`0b0`	TrInProg	Transfer in progress. This field indicates whether a transfer is in progress on the APB master interface.
[6]	UNKNOWN	DeviceEn	Indicates the status of the **ap_en** port. The bit is set when **ap_en** is HIGH, and is clear otherwise. If this bit is clear, no APB transfers are carried out, that is, both Secure and Non-secure accesses are blocked.
[5:4]	`0b00`	AddrInc	Auto address increment mode on RW data access. Only increments if the current transaction completes without an error response and the transaction is not aborted. Reset to `0b0`. **`0x0`** Auto increment OFF **`0x1`** Increment, single. Single transfer from corresponding byte lane **`0x2`** Reserved **`0x3`** Reserved
[3]	`0b0`	RES0	Reserved bit or field with SBZP behavior

Bits	Reset value	Name	Function
[2:0]	`0b010`	Size	Size of the data access to perform. The APB-AP supports only word accesses and this field is fixed at `0x2`. The reset value is `0x2`.

6.3.2.6 Transfer Address Register, TAR

TAR holds the transfer address of the current transfer. TAR must be programmed before initiating any memory transfer through DRW, or Banked Data Registers, or Direct Access Registers.

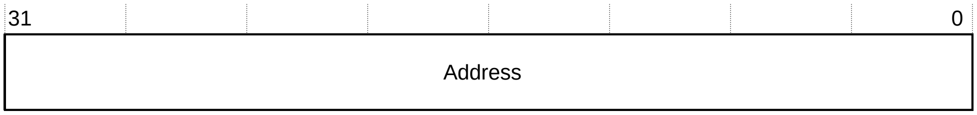
The TAR register characteristics are:

Attributes

Offset	`0x0D04`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-22: TAR register bit assignments



The following table shows the bit assignments.

Table 6-24: TAR register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Address	Address of the current transfer. When a memory access is initiated by accessing the DRW register, the TAR value directly gives the 32-bit transfer address. When a memory access is initiated by accessing Banked Data registers, the TAR only provides the upper bits [31:4] and the remaining address bits [3:0] come from the offset of Banked Data register being accessed. When a memory access is initiated by accessing Direct Access Registers, the TAR provides the upper bits [31:10] and the remaining address bits [9:0] come from the offset of the DAR being accessed.

6.3.2.7 Data Read/Write register, DRW

A write to the DRW register initiates a memory write transaction on the master. AP drives DRW write data on the data bus during the data phase of the current transfer. Reading the DRW register initiates a memory read transaction on the master. The resulting read data that is received from the memory system is returned on the slave interface.

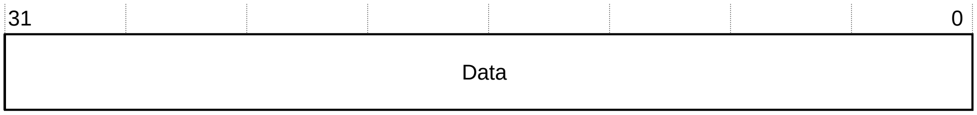
The DRW register characteristics are:

Attributes

Offset	`0x0D0C`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-23: DRW register bit assignments



The following table shows the bit assignments.

Table 6-25: DRW register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Current transfer data value. In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.3.2.8 Banked Data register 0, BD0

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

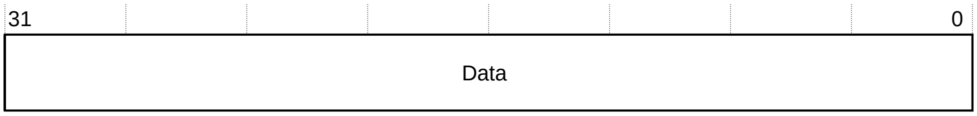
The BD0 register characteristics are:

Attributes

Offset	`0x0D10`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-24: BD0 register bit assignments



The following table shows the bit assignments.

Table 6-26: BD0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFFF0`) + `0x0`). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.3.2.9 Banked Data register 1, BD1

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

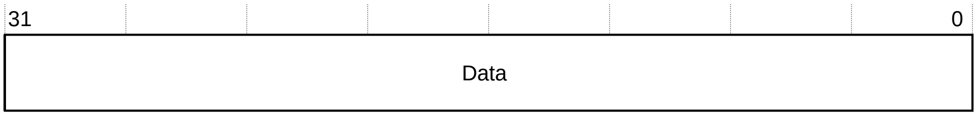
The BD1 register characteristics are:

Attributes

Offset	`0x0D14`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-25: BD1 register bit assignments



The following table shows the bit assignments.

Table 6-27: BD1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFFF0`) + `0x4`). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.3.2.10 Banked Data register 2, BD2

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

The BD2 register characteristics are:

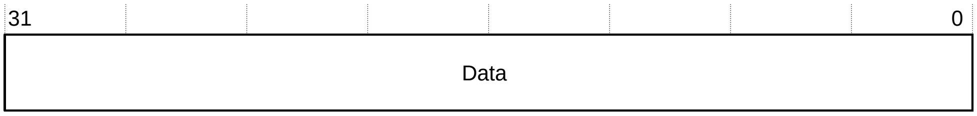
Attributes

Offset	`0x0D18`
Type	Read-write

Reset `0x--------`
Width 32

The following figure shows the bit assignments.

Figure 6-26: BD2 register bit assignments



The following table shows the bit assignments.

Table 6-28: BD2 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFFF0`) + `0x8`). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.3.2.11 Banked Data register 3, BD3

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

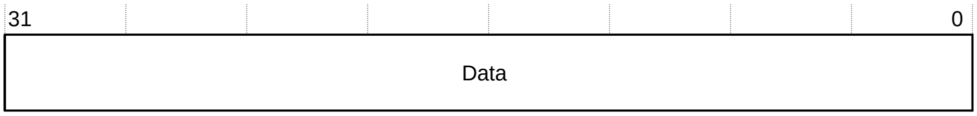
The BD3 register characteristics are:

Attributes

Offset `0x0D1C`
Type Read-write
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

Figure 6-27: BD3 register bit assignments



The following table shows the bit assignments.

Table 6-29: BD3 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFFF0`) + `0xC`). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.3.2.12 Transfer Response Register, TRR

The Transfer Response Register is used to capture an error response received during a transaction. It is also used to clear any logged responses.

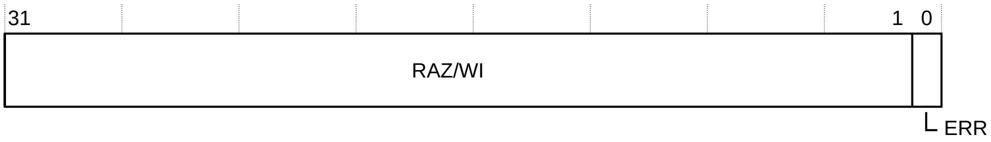
The TRR register characteristics are:

Attributes

Offset	`0x0D24`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-28: TRR register bit assignments



The following table shows the bit assignments.

Table 6-30: TRR register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x0`	RAZ/WI	Read-As-Zero, Writes Ignored
[0]	`0b0`	ERR	Logged error. **0** On reads, no error response logged. Writing to this bit has no effect. **1** On reads, error response logged. Writing to this bit clears this bit to 0.

6.3.2.13 Configuration register, CFG

This is the APBAP Configuration register.

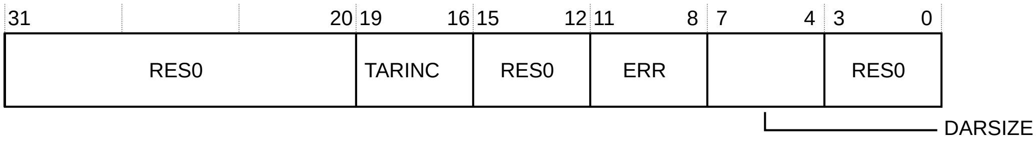
The CFG register characteristics are:

Attributes

Offset `0x0DF4`
Type Read-only
Reset `0x000101A0`
Width 32

The following figure shows the bit assignments.

Figure 6-29: CFG register bit assignments



The following table shows the bit assignments.

Table 6-31: CFG register bit assignments

Bits	Reset value	Name	Function
[31:20]	`0b000000000000`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[19:16]	`0b0001`	TARINC	TAR incrementer size. Returns `0x1` indicating a TAR incrementer size of 10-bits.
[15:12]	`0b0000`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[11:8]	`0b0001`	ERR	Indicates the type of error handling that is implemented. **`0x0`** Error response handling 0. This means that CSW.ERRNPASS, CSW.ERRSTOP, and TRR are not implemented. **`0x1`** Error response handling 1. This means that CSW.ERRNPASS, CSW.ERRSTOP, and TRR are implemented.
[7:4]	`0b1010`	DARSIZE	Size of DAR register space. Returns `0xA` indicating that 1KB (256 registers, each 32-bit wide) of DAR is implemented.
[3:0]	`0b0000`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior

6.3.2.14 Debug Base Address register, BASE

Provides an initial system address for the first component in the system. Typically, the system address is the address of a top-level

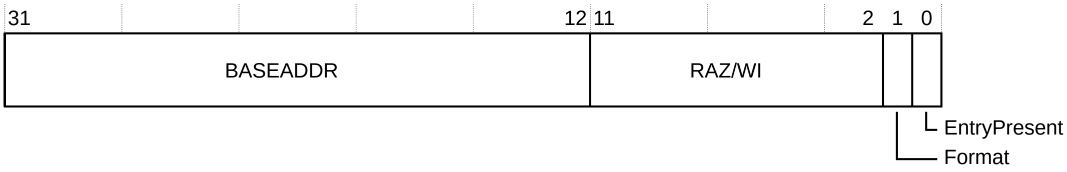
The BASE register characteristics are:

Attributes

Offset `0x0DF8`
Type Read-only
Reset `0x-----00-`
Width 32

The following figure shows the bit assignments.

Figure 6-30: BASE register bit assignments



The following table shows the bit assignments.

Table 6-32: BASE register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION DEFINED	BASEADDR	Base address of a ROM table. It points to the start of the debug register space or a ROM table address. Bits[11:0] of the address are `0x000` because the address is aligned to 4KB boundary. This field is valid only if BASE.EntryPresent bit is set to 1, in which case it returns the tie-off value of the input signal **baseaddr[31:12]**, otherwise, it reads as `0x0`.
[11:2]	`0b0000000000`	RAZ/WI	Read-As-Zero, Writes Ignored
[1]	`0b1`	Format	Base address register format. Returns the value `0b1` indicating the ADIv5 format, which is unchanged in ADIv6.
[0]	IMPLEMENTATION DEFINED	EntryPresent	This field indicates whether a debug component is present for this AP. It returns the tie-off value of the input signal **baseaddr_valid**. **0** No debug entry present **1** Debug entry present and BASE.BASEADDR indicate the start address of the debug register space or ROM table

6.3.2.15 Identification Register, IDR

The IDR provides a mechanism for the debugger to know various identity attributes of the AP.

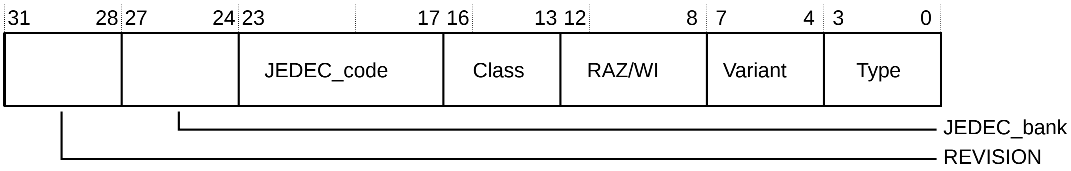
The IDR register characteristics are:

Attributes

Offset	`0x0DFC`
Type	Read-only
Reset	`0x34770006`
Width	32

The following figure shows the bit assignments.

Figure 6-31: IDR register bit assignments



The following table shows the bit assignments.

Table 6-33: IDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0011`	REVISION	Revision. An incremental value starting at `0x0` for the first design of a component. See the Component list in Chapter 1 for information on the RTL revision of the component.
[27:24]	`0b0100`	JEDEC_bank	The JEP106 continuation code. Returns `0x4`, indicating Arm as the designer.
[23:17]	`0b0111011`	JEDEC_code	The JEP106 identification code. Returns `0x3B`, indicating Arm as the designer.
[16:13]	`0b1000`	Class	Returns `0x8`, indicating that this is a MEM-AP
[12:8]	`0b00000`	RAZ/WI	Read-As-Zero, Writes Ignored
[7:4]	`0b0000`	Variant	Returns `0x0`, indicating no variation from base type specified by IDR.Type
[3:0]	`0b0110`	Type	Returns `0x6`, indicating that this is an APB4 Access Port

6.3.2.16 Integration Test Status register, ITSTATUS

Indicates the Integration Test DP Abort status.

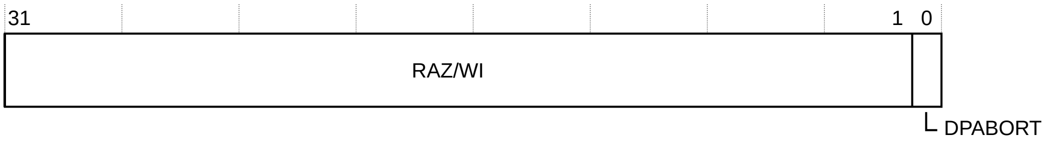
The ITSTATUS register characteristics are:

Attributes

Offset	`0x0EFC`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-32: ITSTATUS register bit assignments



The following table shows the bit assignments.

Table 6-34: ITSTATUS register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x0`	RAZ/WI	Read-As-Zero, Writes Ignored
[0]	`0b0`	DPABORT	When in Integration testing mode (ITCTRL.IME=`0b1`): Behaves as a sticky bit and latches to 1 on a rising edge of **dp_abort**. Cleared on a read from this register. If **dp_abort** rises in the same cycle as a read of the ITSTATUS register is received, the read takes priority and the register is cleared. When in normal functional operation mode (ITCTRL.IME=`0b0`): Read as 0, writes ignored.

6.3.2.17 Integration Mode Control Register, ITCTRL

The Integration Mode Control register is used to enable topology detection.

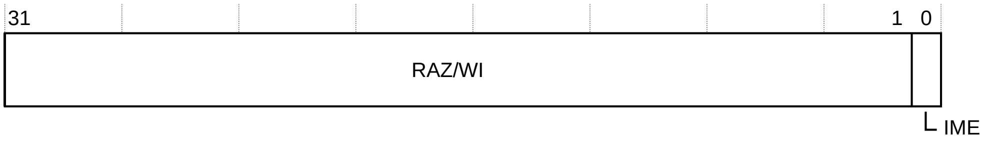
The ITCTRL register characteristics are:

Attributes

Offset	`0x0F00`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-33: ITCTRL register bit assignments



The following table shows the bit assignments.

Table 6-35: ITCTRL register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x0`	RAZ/ WI	Read-As-Zero, Writes Ignored
[0]	`0b0`	IME	Integration Mode Enable. When set, the component enters integration mode, enabling topology detection or integration testing to be performed.

6.3.2.18 Claim Tag Set Register, CLAIMSET

This register forms one half of the claim tag value. On writes, this location enables individual bits to be set. On reads, it returns the number of bits that can be set.

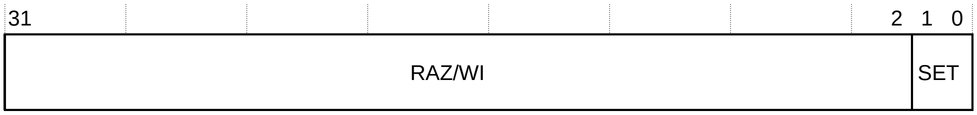
The CLAIMSET register characteristics are:

Attributes

Offset	`0x0FA0`
Type	Read-write
Reset	`0x00000003`
Width	32

The following figure shows the bit assignments.

Figure 6-34: CLAIMSET register bit assignments



The following table shows the bit assignments.

Table 6-36: CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[31:2]	`0x0`	RAZ/WI	Read-As-Zero, Writes Ignored
[1:0]	`0b11`	SET	A bit-programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

6.3.2.19 Claim Tag Clear Register, CLAIMCLR

This register forms one half of the claim tag value. On writes, this location enables individual bits to be cleared. On reads, it returns the current claim tag value.

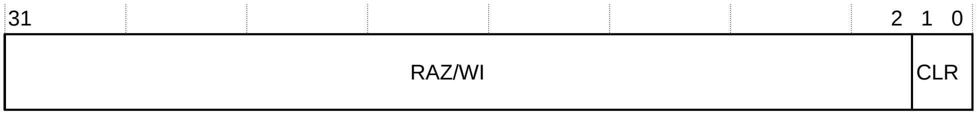
The CLAIMCLR register characteristics are:

Attributes

Offset	`0x0FA4`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-35: CLAIMCLR register bit assignments



The following table shows the bit assignments.

Table 6-37: CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[31:2]	`0x0`	RAZ/WI	Read-As-Zero, Writes Ignored
[1:0]	`0b00`	CLR	A bit-programmable register bank that clears the claim tag value. It is zero at reset. It is used by software agents to signal to each other ownership of the hardware. It has no direct effect on the hardware itself.

6.3.2.20 Authentication Status Register, AUTHSTATUS

Reports the current status of the authentication control signals.

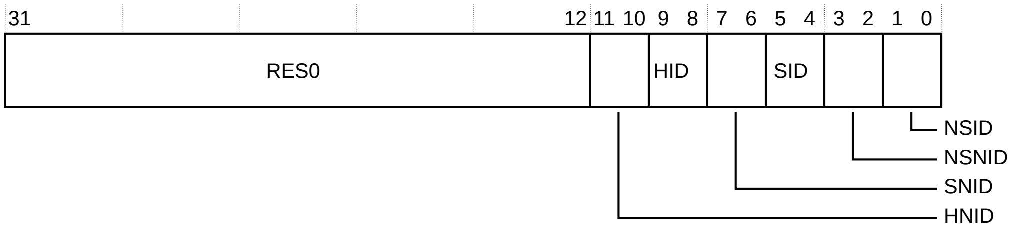
The AUTHSTATUS register characteristics are:

Attributes

Offset	`0x0FB8`
Type	Read-only
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

Figure 6-36: AUTHSTATUS register bit assignments



The following table shows the bit assignments.

Table 6-38: AUTHSTATUS register bit assignments

Bits	Reset value	Name	Function
[31:12]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[11:10]	`0b00`	HNID	Hypervisor non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[9:8]	`0b00`	HID	Hypervisor invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[7:6]	UNKNOWN	SNID	Secure non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[5:4]	UNKNOWN	SID	Secure invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled

Bits	Reset value	Name	Function
[3:2]	UNKNOWN	NSNID	Non-secure non-invasive debug: 0x0 Functionality not implemented or controlled elsewhere 0x1 Reserved 0x2 Functionality disabled 0x3 Functionality enabled
[1:0]	UNKNOWN	NSID	Non-secure invasive debug: 0x0 Functionality not implemented or controlled elsewhere 0x1 Reserved 0x2 Functionality disabled 0x3 Functionality enabled

6.3.2.21 Device Architecture Register, DEVARCH

Identifies the architect and architecture of a CoreSight component. The architect might differ from the designer of a component, for example Arm defines the architecture but another company designs and implements the component.

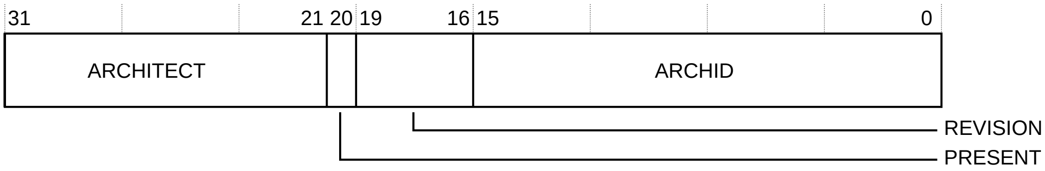
The DEVARCH register characteristics are:

Attributes

Offset	0x0FBC
Type	Read-only
Reset	0x47700A17
Width	32

The following figure shows the bit assignments.

Figure 6-37: DEVARCH register bit assignments



The following table shows the bit assignments.

Table 6-39: DEVARCH register bit assignments

Bits	Reset value	Name	Function
[31:21]	0b01000111011	ARCHITECT	Returns 0x23B, denoting Arm as architect of the component
[20]	0b1	PRESENT	Returns 1, indicating that the DEVARCH register is present
[19:16]	0b0000	REVISION	Architecture revision. Returns the revision of the architecture that the ARCHID field specifies.
[15:0]	0xA17	ARCHID	Architecture ID. Returns 0x0A17, identifying APv2 MEM-AP architecture v0.

6.3.2.22 Device Type Identifier Register, DEVTYPE

A debugger can use this register to get information about a component that has an unrecognized Part number.

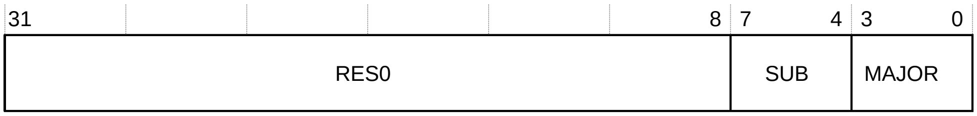
The DEVTYPE register characteristics are:

Attributes

Offset	`0x0FCC`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-38: DEVTYPE register bit assignments



The following table shows the bit assignments.

Table 6-40: DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b0000`	SUB	Minor classification. Returns `0x0`, Other/undefined.
[3:0]	`0b0000`	MAJOR	Major classification. Returns `0x0`, Miscellaneous.

6.3.2.23 Peripheral Identification Register 4, PIDR4

The PIDR4 register is part of the set of peripheral identification registers.

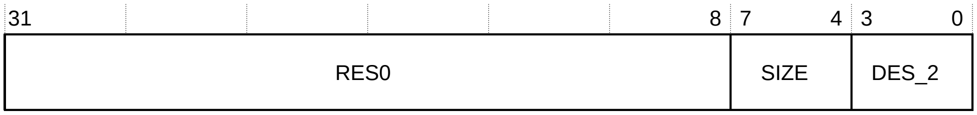
The PIDR4 register characteristics are:

Attributes

Offset	`0x0FD0`
Type	Read-only
Reset	`0x00000004`
Width	32

The following figure shows the bit assignments.

Figure 6-39: PIDR4 register bit assignments



The following table shows the bit assignments.

Table 6-41: PIDR4 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b0000`	SIZE	Indicates the memory size that is used by this component. Returns 0 indicating that the component uses an **UNKNOWN** number of 4KB blocks. Using the SIZE field to indicate the size of the component is deprecated.
[3:0]	`0b0100`	DES_2	JEP106 continuation code. Together, with PIDR2.DES_1 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same.

6.3.2.24 Peripheral Identification Register 5, PIDR5

The PIDR5 register is part of the set of peripheral identification registers.

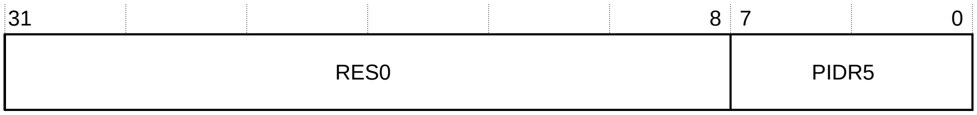
The PIDR5 register characteristics are:

Attributes

Offset	`0x0FD4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-40: PIDR5 register bit assignments



The following table shows the bit assignments.

Table 6-42: PIDR5 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000000`	PIDR5	Reserved

6.3.2.25 Peripheral Identification Register 6, PIDR6

The PIDR6 register is part of the set of peripheral identification registers.

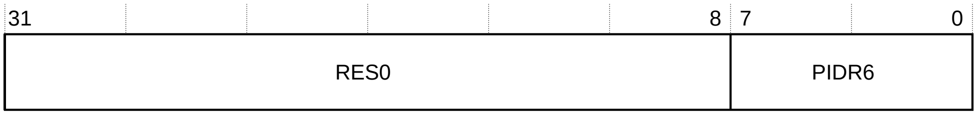
The PIDR6 register characteristics are:

Attributes

Offset	`0x0FD8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-41: PIDR6 register bit assignments



The following table shows the bit assignments.

Table 6-43: PIDR6 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000000`	PIDR6	Reserved

6.3.2.26 Peripheral Identification Register 7, PIDR7

The PIDR7 register is part of the set of peripheral identification registers.

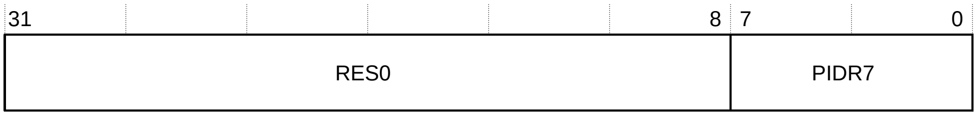
The PIDR7 register characteristics are:

Attributes

Offset	`0x0FDC`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-42: PIDR7 register bit assignments



The following table shows the bit assignments.

Table 6-44: PIDR7 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000000`	PIDR7	Reserved

6.3.2.27 Peripheral Identification Register 0, PIDR0

The PIDR0 register is part of the set of peripheral identification registers.

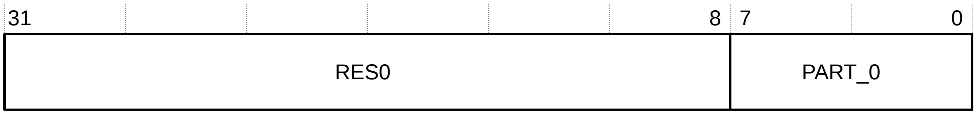
The PIDR0 register characteristics are:

Attributes

Offset	`0x0FE0`
Type	Read-only
Reset	`0x000000E2`
Width	32

The following figure shows the bit assignments.

Figure 6-43: PIDR0 register bit assignments



The following table shows the bit assignments.

Table 6-45: PIDR0 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b11100010`	PART_0	Part number, bits[7:0]. Taken together with PIDR1.PART_1 it indicates the component. The Part Number is selected by the designer of the component.

6.3.2.28 Peripheral Identification Register 1, PIDR1

The PIDR1 register is part of the set of peripheral identification registers.

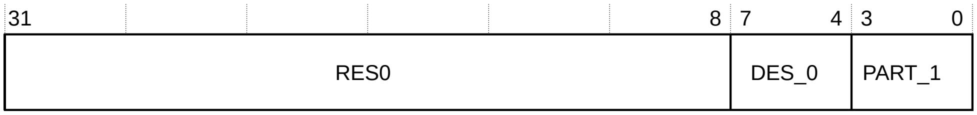
The PIDR1 register characteristics are:

Attributes

Offset	`0x0FE4`
Type	Read-only
Reset	`0x000000B9`
Width	32

The following figure shows the bit assignments.

Figure 6-44: PIDR1 register bit assignments



The following table shows the bit assignments.

Table 6-46: PIDR1 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b1011`	DES_0	JEP106 identification code, bits[3:0]. Together, with PIDR4.DES_2 and PIDR2.DES_1, they indicate the designer of the component and not the implementer, except where the two are the same.
[3:0]	`0b1001`	PART_1	Part number, bits[11:8]. Taken together with PIDR0.PART_0 it indicates the component. The Part Number is selected by the designer of the component.

6.3.2.29 Peripheral Identification Register 2, PIDR2

The PIDR2 register is part of the set of peripheral identification registers.

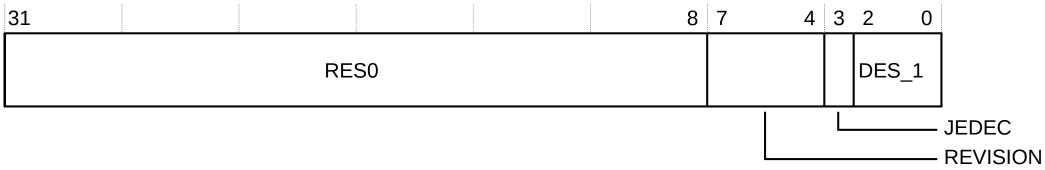
The PIDR2 register characteristics are:

Attributes

Offset	`0x0FE8`
Type	Read-only
Reset	`0x0000003B`
Width	32

The following figure shows the bit assignments.

Figure 6-45: PIDR2 register bit assignments



The following table shows the bit assignments.

Table 6-47: PIDR2 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b0011`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the Component list in Chapter 1 for information on the RTL revision of the component.

Bits	Reset value	Name	Function
[3]	`0b1`	JEDEC	1 - Always set. Indicates that a JEDEC assigned value is used.
[2:0]	`0b011`	DES_1	JEP106 identification code, bits[6:4]. Together, with PIDR4.DES_2 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same.

6.3.2.30 Peripheral Identification Register 3, PIDR3

The PIDR3 register is part of the set of peripheral identification registers.

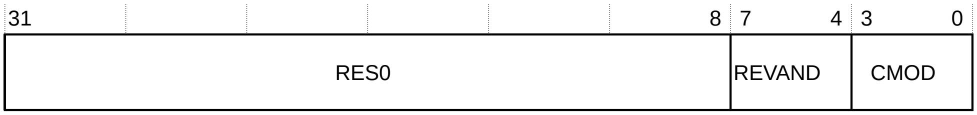
The PIDR3 register characteristics are:

Attributes

Offset	`0x0FEC`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-46: PIDR3 register bit assignments



The following table shows the bit assignments.

Table 6-48: PIDR3 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b0000`	REVAND	This field indicates minor errata fixes specific to this design, for example metal fixes after implementation. In most cases this field is `0x0`.
[3:0]	`0b0000`	CMOD	Customer Modified. Where the component is reusable IP, this value indicates if the customer has modified the behavior of the component. In most cases this field is `0x0`.

6.3.2.31 Component Identification Register 0, CIDR0

The CIDR0 register is part of the set of component identification registers.

The CIDR0 register characteristics are:

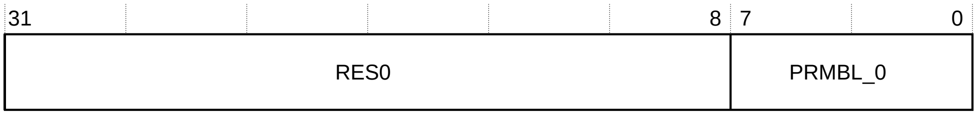
Attributes

Offset	`0x0FF0`
Type	Read-only
Reset	`0x0000000D`

Width 32

The following figure shows the bit assignments.

Figure 6-47: CIDR0 register bit assignments



The following table shows the bit assignments.

Table 6-49: CIDR0 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

6.3.2.32 Component Identification Register 1, CIDR1

The CIDR1 register is part of the set of component identification registers.

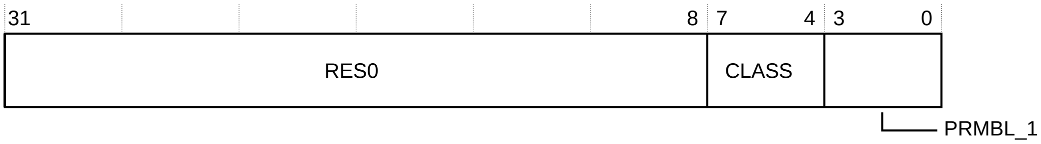
The CIDR1 register characteristics are:

Attributes

Offset	`0x0FF4`
Type	Read-only
Reset	`0x00000090`
Width	32

The following figure shows the bit assignments.

Figure 6-48: CIDR1 register bit assignments



The following table shows the bit assignments.

Table 6-50: CIDR1 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

6.3.2.33 Component Identification Register 2, CIDR2

The CIDR2 register is part of the set of component identification registers.

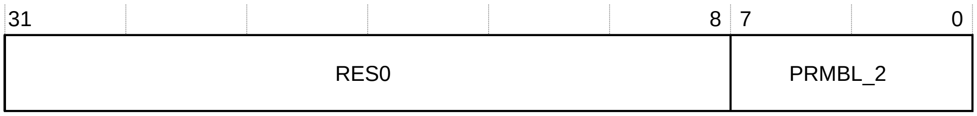
The CIDR2 register characteristics are:

Attributes

Offset	`0x0FF8`
Type	Read-only
Reset	`0x00000005`
Width	32

The following figure shows the bit assignments.

Figure 6-49: CIDR2 register bit assignments



The following table shows the bit assignments.

Table 6-51: CIDR2 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

6.3.2.34 Component Identification Register 3, CIDR3

The CIDR3 register is part of the set of component identification registers.

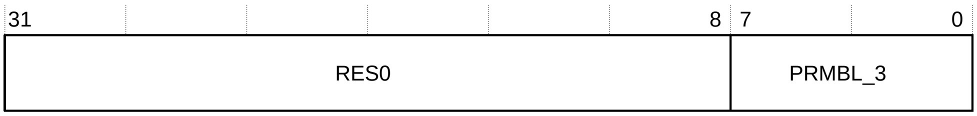
The CIDR3 register characteristics are:

Attributes

Offset	`0x0FFC`
Type	Read-only
Reset	`0x000000B1`
Width	32

The following figure shows the bit assignments.

Figure 6-50: CIDR3 register bit assignments



The following table shows the bit assignments.

Table 6-52: CIDR3 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

6.4 css600_ahbap introduction

This section describes the programmers model of the `css600_ahbap`.

6.4.1 Register summary

The following table shows the registers in offset order from the base memory address.

Note

A reset value containing one or more '-' means that this register contains **UNKNOWN** or **IMPLEMENTATION-DEFINED** values. See the relevant register description for more information.

Locations that are not listed in the table are Reserved.

The 8KB memory map contains two views of the registers, one starting at `0x00000000`, and the other at `0x00001000`.

Only 4KB can be accessed as there is no twin AP.

In the case of RW registers, the two views provide independent physical registers. Writing to a RW register in one view does not affect the contents of the same register in the other view. For all read-only registers, the two views provide read access to the same physical register. In this case, reading from either view results in the same data being read.

Table 6-53: `css600_ahbap - APB4_Slave_0` register summary

Offset	Name	Type	Reset	Width	Description
`0x0000`	DAR0	RW	`0x--------`	32	6.4.2.1 Direct Access Register 0, DAR0 on page 79
`0x0004`	DAR1	RW	`0x--------`	32	6.4.2.2 Direct Access Register 1, DAR1 on page 80
`0x0008`	DAR2	RW	`0x--------`	32	6.4.2.3 Direct Access Register 2, DAR2 on page 80
…	…	…	…	…	…
`0x03FC`	DAR255	RW	`0x--------`	32	6.4.2.4 Direct Access Register 255, DAR255 on page 81
`0x0D00`	CSW	RW	`0x43-000-2`	32	6.4.2.5 Control Status Word register, CSW on page 82
`0x0D04`	TAR	RW	`0x--------`	32	6.4.2.6 Transfer Address Register, TAR on page 84
`0x0D0C`	DRW	RW	`0x--------`	32	6.4.2.7 Data Read/Write register, DRW on page 85

Offset	Name	Type	Reset	Width	Description
0x0D10	BD0	RW	0x--------	32	6.4.2.8 Banked Data register 0, BD0 on page 85
0x0D14	BD1	RW	0x--------	32	6.4.2.9 Banked Data register 1, BD1 on page 86
0x0D18	BD2	RW	0x--------	32	6.4.2.10 Banked Data register 2, BD2 on page 87
0x0D1C	BD3	RW	0x--------	32	6.4.2.11 Banked Data register 3, BD3 on page 87
0x0D24	TRR	RW	0x00000000	32	6.4.2.12 Transfer Response Register, TRR on page 88
0x0DF4	CFG	RO	0x000101A0	32	6.4.2.13 Configuration register, CFG on page 89
0x0DF8	BASE	RO	0x-----00-	32	6.4.2.14 Debug Base Address register, BASE on page 89
0x0DFC	IDR	RO	0x44770008	32	6.4.2.15 Identification Register, IDR on page 90
0x0EFC	ITSTATUS	RW	0x00000000	32	6.4.2.16 Integration Test Status register, ITSTATUS on page 91
0x0F00	ITCTRL	RW	0x00000000	32	6.4.2.17 Integration Mode Control Register, ITCTRL on page 92
0x0FA0	CLAIMSET	RW	0x00000003	32	6.4.2.18 Claim Tag Set Register, CLAIMSET on page 92
0x0FA4	CLAIMCLR	RW	0x00000000	32	6.4.2.19 Claim Tag Clear Register, CLAIMCLR on page 93
0x0FB8	AUTHSTATUS	RO	0x000000--	32	6.4.2.20 Authentication Status Register, AUTHSTATUS on page 94
0x0FBC	DEVARCH	RO	0x47700A17	32	6.4.2.21 Device Architecture Register, DEVARCH on page 95
0x0FCC	DEVTYPE	RO	0x00000000	32	6.4.2.22 Device Type Identifier Register, DEVTYPE on page 96
0x0FD0	PIDR4	RO	0x00000004	32	6.4.2.23 Peripheral Identification Register 4, PIDR4 on page 97
0x0FD4	PIDR5	RO	0x00000000	32	6.4.2.24 Peripheral Identification Register 5, PIDR5 on page 97
0x0FD8	PIDR6	RO	0x00000000	32	6.4.2.25 Peripheral Identification Register 6, PIDR6 on page 98
0x0FDC	PIDR7	RO	0x00000000	32	6.4.2.26 Peripheral Identification Register 7, PIDR7 on page 98
0x0FE0	PIDR0	RO	0x000000E3	32	6.4.2.27 Peripheral Identification Register 0, PIDR0 on page 99
0x0FE4	PIDR1	RO	0x000000B9	32	6.4.2.28 Peripheral Identification Register 1, PIDR1 on page 100
0x0FE8	PIDR2	RO	0x0000004B	32	6.4.2.29 Peripheral Identification Register 2, PIDR2 on page 100
0x0FEC	PIDR3	RO	0x00000000	32	6.4.2.30 Peripheral Identification Register 3, PIDR3 on page 101
0x0FF0	CIDR0	RO	0x0000000D	32	6.4.2.31 Component Identification Register 0, CIDR0 on page 102
0x0FF4	CIDR1	RO	0x00000090	32	6.4.2.32 Component Identification Register 1, CIDR1 on page 102
0x0FF8	CIDR2	RO	0x00000005	32	6.4.2.33 Component Identification Register 2, CIDR2 on page 103
0x0FFC	CIDR3	RO	0x000000B1	32	6.4.2.34 Component Identification Register 3, CIDR3 on page 104
0x1000	DAR0	RW	0x--------	32	6.4.2.1 Direct Access Register 0, DAR0 on page 79
0x1004	DAR1	RW	0x--------	32	6.4.2.2 Direct Access Register 1, DAR1 on page 80
0x1008	DAR2	RW	0x--------	32	6.4.2.3 Direct Access Register 2, DAR2 on page 80
...	...	...	...	...	...
0x13FC	DAR255	RW	0x--------	32	6.4.2.4 Direct Access Register 255, DAR255 on page 81
0x1D00	CSW	RW	0x43-000-2	32	6.4.2.5 Control Status Word register, CSW on page 82
0x1D04	TAR	RW	0x--------	32	6.4.2.6 Transfer Address Register, TAR on page 84
0x1D0C	DRW	RW	0x--------	32	6.4.2.7 Data Read/Write register, DRW on page 85
0x1D10	BD0	RW	0x--------	32	6.4.2.8 Banked Data register 0, BD0 on page 85
0x1D14	BD1	RW	0x--------	32	6.4.2.9 Banked Data register 1, BD1 on page 86
0x1D18	BD2	RW	0x--------	32	6.4.2.10 Banked Data register 2, BD2 on page 87
0x1D1C	BD3	RW	0x--------	32	6.4.2.11 Banked Data register 3, BD3 on page 87
0x1D24	TRR	RW	0x00000000	32	6.4.2.12 Transfer Response Register, TRR on page 88

Offset	Name	Type	Reset	Width	Description
`0x1DF4`	CFG	RO	`0x000101A0`	32	6.4.2.13 Configuration register, CFG on page 89
`0x1DF8`	BASE	RO	`0x-----00-`	32	6.4.2.14 Debug Base Address register, BASE on page 89
`0x1DFC`	IDR	RO	`0x44770008`	32	6.4.2.15 Identification Register, IDR on page 90
`0x1EFC`	ITSTATUS	RW	`0x00000000`	32	6.4.2.16 Integration Test Status register, ITSTATUS on page 91
`0x1F00`	ITCTRL	RW	`0x00000000`	32	6.4.2.17 Integration Mode Control Register, ITCTRL on page 92
`0x1FA0`	CLAIMSET	RW	`0x00000003`	32	6.4.2.18 Claim Tag Set Register, CLAIMSET on page 92
`0x1FA4`	CLAIMCLR	RW	`0x00000000`	32	6.4.2.19 Claim Tag Clear Register, CLAIMCLR on page 93
`0x1FB8`	AUTHSTATUS	RO	`0x000000--`	32	6.4.2.20 Authentication Status Register, AUTHSTATUS on page 94
`0x1FBC`	DEVARCH	RO	`0x47700A17`	32	6.4.2.21 Device Architecture Register, DEVARCH on page 95
`0x1FCC`	DEVTYPE	RO	`0x00000000`	32	6.4.2.22 Device Type Identifier Register, DEVTYPE on page 96
`0x1FD0`	PIDR4	RO	`0x00000004`	32	6.4.2.23 Peripheral Identification Register 4, PIDR4 on page 97
`0x1FD4`	PIDR5	RO	`0x00000000`	32	6.4.2.24 Peripheral Identification Register 5, PIDR5 on page 97
`0x1FD8`	PIDR6	RO	`0x00000000`	32	6.4.2.25 Peripheral Identification Register 6, PIDR6 on page 98
`0x1FDC`	PIDR7	RO	`0x00000000`	32	6.4.2.26 Peripheral Identification Register 7, PIDR7 on page 98
`0x1FE0`	PIDR0	RO	`0x000000E3`	32	6.4.2.27 Peripheral Identification Register 0, PIDR0 on page 99
`0x1FE4`	PIDR1	RO	`0x000000B9`	32	6.4.2.28 Peripheral Identification Register 1, PIDR1 on page 100
`0x1FE8`	PIDR2	RO	`0x0000004B`	32	6.4.2.29 Peripheral Identification Register 2, PIDR2 on page 100
`0x1FEC`	PIDR3	RO	`0x00000000`	32	6.4.2.30 Peripheral Identification Register 3, PIDR3 on page 101
`0x1FF0`	CIDR0	RO	`0x0000000D`	32	6.4.2.31 Component Identification Register 0, CIDR0 on page 102
`0x1FF4`	CIDR1	RO	`0x00000090`	32	6.4.2.32 Component Identification Register 1, CIDR1 on page 102
`0x1FF8`	CIDR2	RO	`0x00000005`	32	6.4.2.33 Component Identification Register 2, CIDR2 on page 103
`0x1FFC`	CIDR3	RO	`0x000000B1`	32	6.4.2.34 Component Identification Register 3, CIDR3 on page 104

6.4.2 Register descriptions

This section describes the `css600_ahbap` registers.

6.4.1 css600_ahbap - APB4_Slave_0 register summary on page 77 provides cross references to individual registers.

6.4.2.1 Direct Access Register 0, DAR0

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

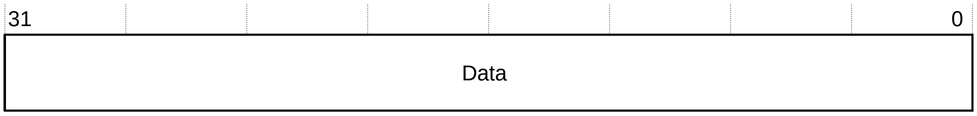
The DAR0 register characteristics are:

Attributes

Offset	`0x0000`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-51: DAR0 register bit assignments



The following table shows the bit assignments.

Table 6-54: DAR0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x0`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.4.2.2 Direct Access Register 1, DAR1

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

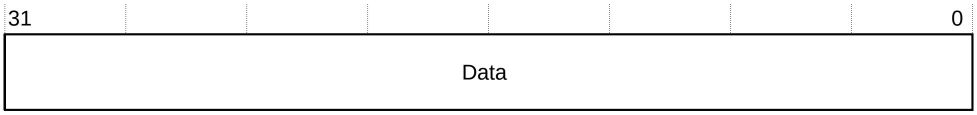
The DAR1 register characteristics are:

Attributes

Offset	`0x0004`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-52: DAR1 register bit assignments



The following table shows the bit assignments.

Table 6-55: DAR1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x4`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.4.2.3 Direct Access Register 2, DAR2

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

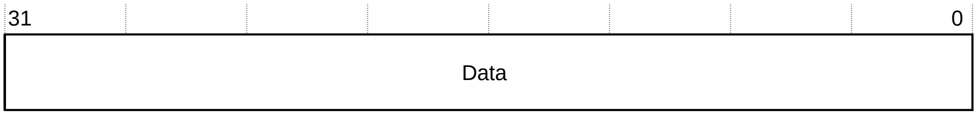
The DAR2 register characteristics are:

Attributes

Offset	`0x0008`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-53: DAR2 register bit assignments



The following table shows the bit assignments.

Table 6-56: DAR2 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x8`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.4.2.4 Direct Access Register 255, DAR255

The Direct Access Registers provide a mechanism for directly mapping locations in the target memory system that is connected to the APB master interface.

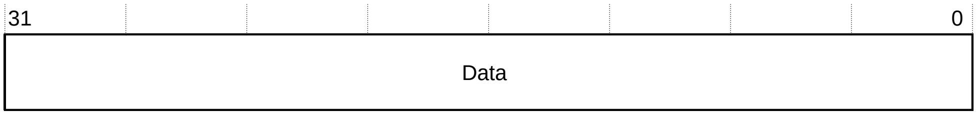
The DAR255 register characteristics are:

Attributes

Offset	`0x03FC`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-54: DAR255 register bit assignments



The following table shows the bit assignments.

Table 6-57: DAR255 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFC00`) + `0x3FC`.) In read mode, the register contains the data value that was read from memory, and in write mode the register contains the data value to write to memory.

6.4.2.5 Control Status Word register, CSW

The CSW register configures and controls accesses through the AHB master interface to the connected memory system.

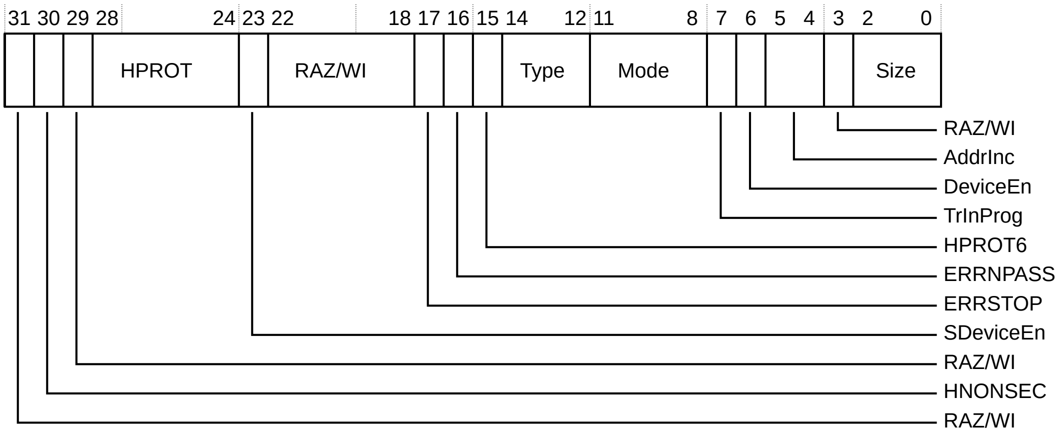
The CSW register characteristics are:

Attributes

Offset	`0x0D00`
Type	Read-write
Reset	`0x43-000-2`
Width	32

The following figure shows the bit assignments.

Figure 6-55: CSW register bit assignments



The following table shows the bit assignments.

Table 6-58: CSW register bit assignments

Bits	Reset value	Name	Function
[31]	`0b0`	RAZ/WI	RAZ/WI
[30]	`0b1`	HNONSEC	Drives **hnonsec_m** output pin. Together with the Access Port Enable interface signals, HNONSEC determines whether a secure access is allowed on the master interface. Accesses are permitted as follows: access_permitted = (ap_en && ap_secure_en) || (ap_en && HNONSEC).
[29]	`0b0`	RAZ/WI	RAZ/WI

[28:24]	`0b00011`	HPROT	This field, in combination with CSW.HPROT6, sets the protection control value to be output on **hprot_m[6:0]**. CSW.HPROT6 controls **hprot_m[6]**. CSW.HPROT controls **hprot_m[4:0]**. **hprot_m[5]** is always driven LOW. This field is reset to `0x3`. The reset values of the two fields correspond to a protection value of: Non-Shareable, (Non-Allocate), Non-Lookup, Non-Modifiable, Non-Bufferable, Privileged, Data. css600_ahbap supports the following legal **hprot_m** encodings:
			CSW.HPROT6=`0b0`, CSW.HPROT[4:2]=`0b000`: Device-nE
			CSW.HPROT6=`0b0`, CSW.HPROT[4:2]=`0b001`: Device-E
			CSW.HPROT6=`0b0`, CSW.HPROT[4:2]=`0b010`: Normal Non-cacheable, Non-shareable
			CSW.HPROT6=`0b0`, CSW.HPROT[4:2]=`0b110`: Write-through, Non-shareable
			CSW.HPROT6=`0b0`, CSW.HPROT[4:2]=`0b111`: Write-back, Non-shareable
			CSW.HPROT6=`0b1`, CSW.HPROT[4:2]=`0b010`: Normal Non-cacheable, Shareable
			CSW.HPROT6=`0b1`, CSW.HPROT[4:2]=`0b110`: Write-through, Shareable
			CSW.HPROT6=`0b1`, CSW.HPROT[4:2]=`0b111`: Write-back, Shareable

[23]	UNKNOWN	SDeviceEn	Indicates the status of the **ap_en** and **ap_secure_en** ports. It is set when both **ap_en** and **ap_secure_en** are HIGH, and remains clear otherwise. If this bit is clear, Secure AHB transfers are not permitted. Non-secure memory accesses and internal register accesses that do not initiate memory accesses are permitted regardless of the status of this bit.
[22:18]	`0b00000`	RAZ/WI	RAZ/WI
[17]	`0b0`	ERRSTOP	Stop on error. **0** Memory access errors do not prevent future memory accesses. **1** Memory access errors prevent future memory accesses.
[16]	`0b0`	ERRNPASS	Errors are not passed upstream. **0** Memory access errors are passed upstream. **1** Memory access errors are not passed upstream.
[15]	`0b0`	HPROT6	This field drives **hprot_s[6]**. Reset to 0. In combination with CSW.HPROT, controls the protection value to be output on **hprot_m[6:0]**. This field is reset to `0x0`.
[14:12]	`0b000`	Type	This field is reserved. Reads return `0x0` and writes are ignored.
[11:8]	`0b0000`	Mode	Specifies the mode of operation. All other values are reserved. **`0x0`** Normal download or upload mode.
[7]	`0b0`	TrInProg	Transfer in progress. This field indicates whether a transfer is in progress on the AHB master interface. If the master interface is busy, CSW.TrInProg is set in both logical APs.

Bits	Reset value	Name	Function
[6]	UNKNOWN	DeviceEn	Indicates the status of the **ap_en** port. The bit is set when **ap_en** is HIGH, and is clear otherwise. If this bit is clear, no AHB transfers are carried out, that is, both secure and non-secure accesses are blocked.
[5:4]	`0b00`	AddrInc	Auto address increment mode on RW data access. Only increments if the current transaction completes without an error response and the transaction is not aborted. **`0x0`** Auto increment OFF **`0x1`** Increment, single. Single transfer from corresponding byte lane. **`0x2`** Reserved **`0x3`** Reserved
[3]	`0b0`	RAZ/WI	RAZ/WI
[2:0]	`0b010`	Size	Size of the data access to perform: **`0x0`** 8 bits **`0x1`** 16 bits **`0x2`** 32 bits **`0x3`** Reserved **`0x4`** Reserved **`0x5`** Reserved **`0x6`** Reserved **`0x7`** Reserved

6.4.2.6 Transfer Address Register, TAR

TAR holds the transfer address of the current transfer. TAR must be programmed before initiating any memory transfer through DRW, or Banked Data Registers, or Direct Access Registers.

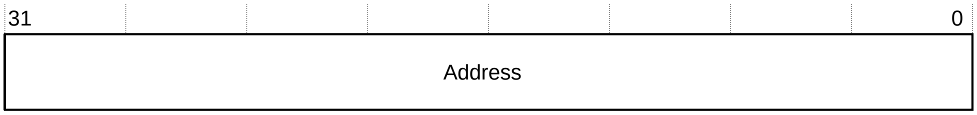
The TAR register characteristics are:

Attributes

Offset	`0x0D04`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-56: TAR register bit assignments



The following table shows the bit assignments.

Table 6-59: TAR register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Address	Address of the current transfer. When a memory access is initiated by accessing the DRW register, the TAR value directly gives the 32-bit transfer address. When a memory access is initiated by accessing Banked Data registers, the TAR only provides the upper bits [31:4] and the remaining address bits [3:0] come from the offset of Banked Data register being accessed. When a memory access is initiated by accessing Direct Access Registers, the TAR provides the upper bits [31:10] and the remaining address bits [9:0] come from the offset of the DAR being accessed.

6.4.2.7 Data Read/Write register, DRW

A write to the DRW register initiates a memory write transaction on the master. AP drives DRW write data on the data bus during the data phase of the current transfer. Reading the DRW register initiates a memory read transaction on the master. The resulting read data that is received from the memory system is returned on the slave interface.

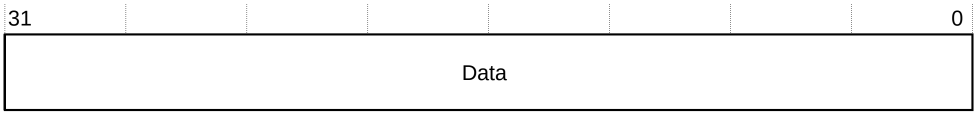
The DRW register characteristics are:

Attributes

Offset	`0x0D0C`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-57: DRW register bit assignments



The following table shows the bit assignments.

Table 6-60: DRW register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Current transfer data value. In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.4.2.8 Banked Data register 0, BD0

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

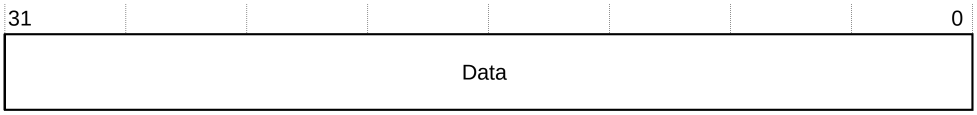
The BD0 register characteristics are:

Attributes

Offset	`0x0D10`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-58: BD0 register bit assignments



The following table shows the bit assignments.

Table 6-61: BD0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFFF0`) + `0x0`). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.4.2.9 Banked Data register 1, BD1

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

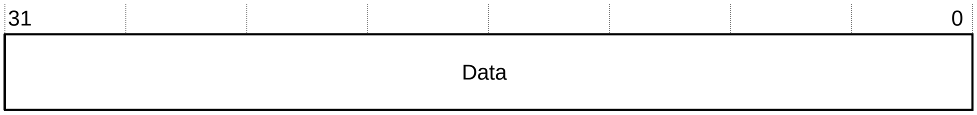
The BD1 register characteristics are:

Attributes

Offset	`0x0D14`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-59: BD1 register bit assignments



The following table shows the bit assignments.

Table 6-62: BD1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & 0xFFFFFFF0) + 0x4). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.4.2.10 Banked Data register 2, BD2

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

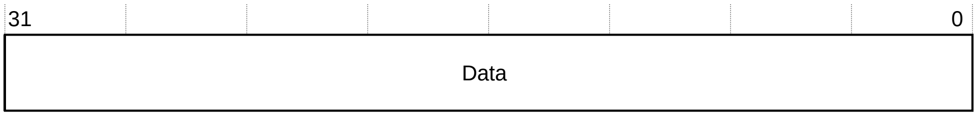
The BD2 register characteristics are:

Attributes

Offset	0x0D18
Type	Read-write
Reset	0x--------
Width	32

The following figure shows the bit assignments.

Figure 6-60: BD2 register bit assignments



The following table shows the bit assignments.

Table 6-63: BD2 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & 0xFFFFFFF0) + 0x8). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.4.2.11 Banked Data register 3, BD3

The Banked Data registers provide a mechanism for directly mapping APB slave accesses to memory transfers without having to rewrite the TAR within a 16-byte boundary.

The BD3 register characteristics are:

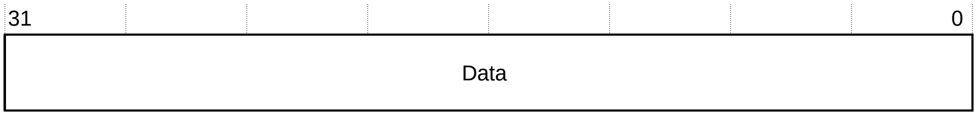
Attributes

Offset	0x0D1C
Type	Read-write

Reset `0x--------`
Width 32

The following figure shows the bit assignments.

Figure 6-61: BD3 register bit assignments



The following table shows the bit assignments.

Table 6-64: BD3 register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	Data	Maps to memory address ((TAR & `0xFFFFFFF0`) + `0xC`). In read mode, the register contains the data value that was read from the current transfer, and in write mode the register contains the data value to write for the current transfer.

6.4.2.12 Transfer Response Register, TRR

The Transfer Response Register is used to capture an error response received during a transaction. It is also used to clear any logged responses.

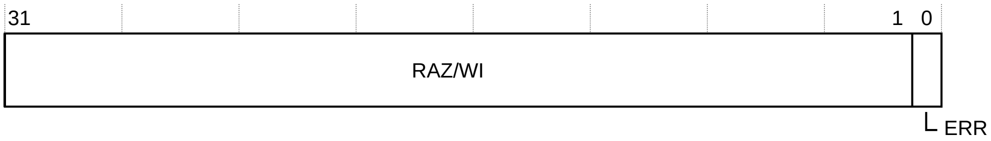
The TRR register characteristics are:

Attributes

Offset `0x0D24`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

Figure 6-62: TRR register bit assignments



The following table shows the bit assignments.

Table 6-65: TRR register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x0`	RAZ/WI	RAZ/WI

Bits	Reset value	Name	Function
[0]	`0b0`	ERR	Logged error: **0** On reads, no error response logged. Writing to this bit has no effect. **1** On reads, error response logged. Writing to this bit clears this bit to 0.

6.4.2.13 Configuration register, CFG

This is the AHBAP Configuration register.

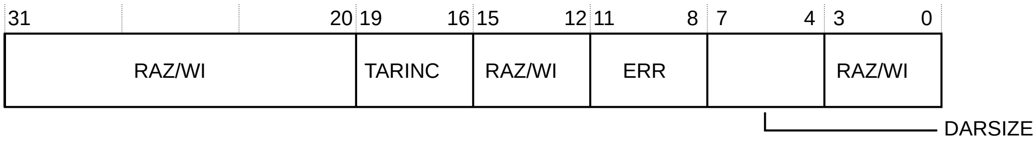
The CFG register characteristics are:

Attributes

Offset	`0x0DF4`
Type	Read-only
Reset	`0x000101A0`
Width	32

The following figure shows the bit assignments.

Figure 6-63: CFG register bit assignments



The following table shows the bit assignments.

Table 6-66: CFG register bit assignments

Bits	Reset value	Name	Function
[31:20]	`0b000000000000`	RAZWI	RAZWI
[19:16]	`0b0001`	TARINC	TAR incrementer size. Returns `0x1` indicating a TAR incrementer size of 10 bits
[15:12]	`0b0000`	RAZWI	RAZWI
[11:8]	`0b0001`	ERR	Error functionality implemented. Returns `0x1` indicating that Error Response Handling version 1 is implemented. See the *Arm Debug Interface Architecture Specification ADIv6.0* for more information.
[7:4]	`0b1010`	DARSIZE	Size of DAR register space. Returns `0xA` indicating that 1KB (256 registers, each 32-bit wide) of DAR is implemented.
[3:0]	`0b0000`	RAZWI	RAZWI

6.4.2.14 Debug Base Address register, BASE

Provides an initial system address for the first component in the system. Typically, the system address is the address of a top-level

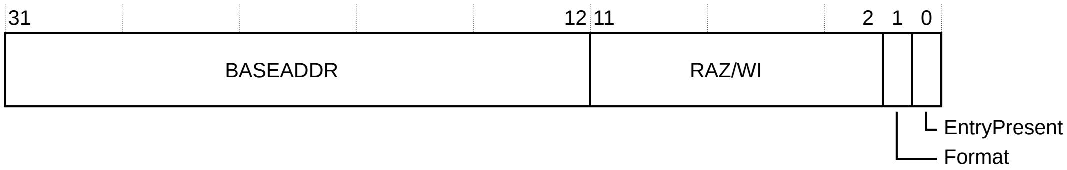
The BASE register characteristics are:

Attributes

Offset	`0x0DF8`
Type	Read-only
Reset	`0x-----00-`
Width	32

The following figure shows the bit assignments.

Figure 6-64: BASE register bit assignments



The following table shows the bit assignments.

Table 6-67: BASE register bit assignments

Bits	Reset value	Name	Function
[31:12]	**IMPLEMENTATION DEFINED**	BASEADDR	Base address of a ROM table. It points to the start of the debug register space or a ROM table address. Bits[11:0] of the address are `0x000` because the address is aligned to 4KB boundary. This field is valid only if BASE.EntryPresent bit is set to 1, in which case it returns the tie-off value of the input signal **baseaddr[31:12]**, otherwise, it reads as `0x0`.
[11:2]	`0b0000000000`	RAZ/WI	RAZ/WI
[1]	`0b1`	Format	Base address register format. Returns the value `0b1` indicating the ADIv5 format, which is unchanged in ADIv6.
[0]	**IMPLEMENTATION DEFINED**	EntryPresent	This field indicates whether a debug component is present for this AP. It returns the tie-off value of the input signal **baseaddr_valid**. **0** No debug entry present **1** Debug entry present and BASE.BASEADDR indicate the start address of the debug register space or ROM table

6.4.2.15 Identification Register, IDR

The IDR provides a mechanism for the debugger to know various identity attributes of the AP.

The IDR register characteristics are:

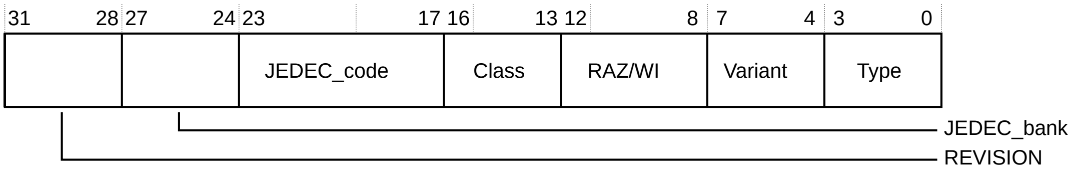
Attributes

Offset	`0x0DFC`
Type	Read-only

Reset `0x44770008`
Width 32

The following figure shows the bit assignments.

Figure 6-65: IDR register bit assignments



The following table shows the bit assignments.

Table 6-68: IDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0100`	REVISION	Revision. An incremental value starting at `0x0` for the first design of a component. See the Component list in Chapter 1 for information on the RTL revision of the component.
[27:24]	`0b0100`	JEDEC_bank	The JEP106 continuation code. Returns `0x4`, indicating Arm as the designer.
[23:17]	`0b0111011`	JEDEC_code	The JEP106 identification code. Returns `0x3B`, indicating Arm as the designer.
[16:13]	`0b1000`	Class	Returns `0x8`, indicating that this is a Memory Access Port
[12:8]	`0b00000`	RAZ/WI	RAZ/WI
[7:4]	`0b0000`	Variant	Returns `0x0`, indicating no variation from base type specified by IDR.Type
[3:0]	`0b1000`	Type	Returns `0x8`, indicating that this is an AHB5 Access Port with full HPROT control

6.4.2.16 Integration Test Status register, ITSTATUS

Indicates the Integration Test DP Abort status.

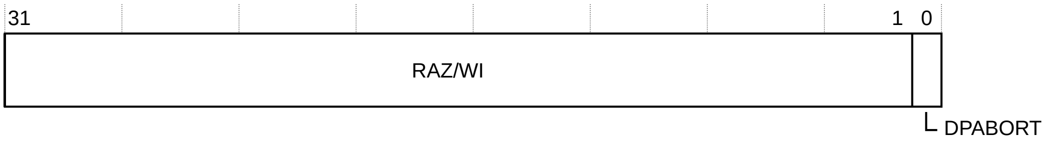
The ITSTATUS register characteristics are:

Attributes

Offset `0x0EFC`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

Figure 6-66: ITSTATUS register bit assignments



The following table shows the bit assignments.

Table 6-69: ITSTATUS register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x0`	RAZ/WI	RAZ/WI
[0]	`0b0`	DPABORT	When in Integration testing mode (ITCTRL.IME=`0b1`): Behaves as a sticky bit and latches to 1 on a rising edge of **dp_abort**. Cleared on a read from this register. If **dp_abort** rises in the same cycle as a read of the ITSTATUS register is received, the read takes priority and the register is cleared. When in normal functional operation mode (ITCTRL.IME=`0b0`): Read as 0, writes ignored.

6.4.2.17 Integration Mode Control Register, ITCTRL

The Integration Mode Control register is used to enable topology detection.

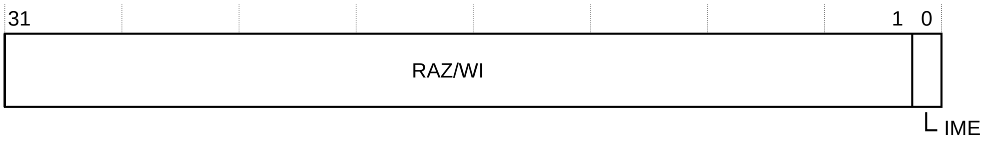
The ITCTRL register characteristics are:

Attributes

Offset	`0x0F00`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-67: ITCTRL register bit assignments



The following table shows the bit assignments.

Table 6-70: ITCTRL register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x0`	RAZ/WI	RAZ/WI
[0]	`0b0`	IME	Integration Mode Enable. When set, the component enters integration mode, enabling topology detection or integration testing to be performed.

6.4.2.18 Claim Tag Set Register, CLAIMSET

This register forms one half of the claim tag value. On writes, this location enables individual bits to be set. On reads, it returns the number of bits that can be set.

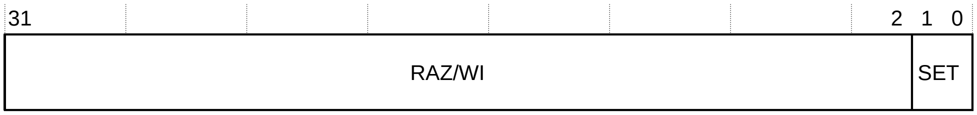
The CLAIMSET register characteristics are:

Attributes

Offset	`0x0FA0`
Type	Read-write
Reset	`0x00000003`
Width	32

The following figure shows the bit assignments.

Figure 6-68: CLAIMSET register bit assignments



The following table shows the bit assignments.

Table 6-71: CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[31:2]	`0x0`	RAZ/ WI	RAZ/WI
[1:0]	`0b11`	SET	A bit-programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

6.4.2.19 Claim Tag Clear Register, CLAIMCLR

This register forms one half of the claim tag value. On writes, this location enables individual bits to be cleared. On reads, it returns the current claim tag value.

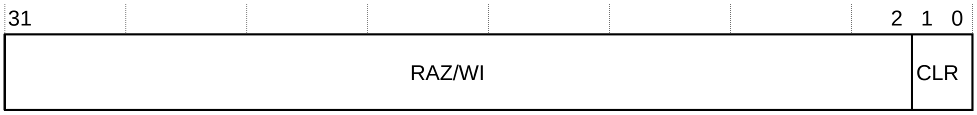
The CLAIMCLR register characteristics are:

Attributes

Offset	`0x0FA4`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-69: CLAIMCLR register bit assignments



The following table shows the bit assignments.

Table 6-72: CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[31:2]	`0x0`	RAZ/ WI	RAZ/WI
[1:0]	`0b00`	CLR	A bit-programmable register bank that clears the claim tag value. It is zero at reset. It is used by software agents to signal to each other ownership of the hardware. It has no direct effect on the hardware itself.

6.4.2.20 Authentication Status Register, AUTHSTATUS

Reports the current status of the authentication control signals.

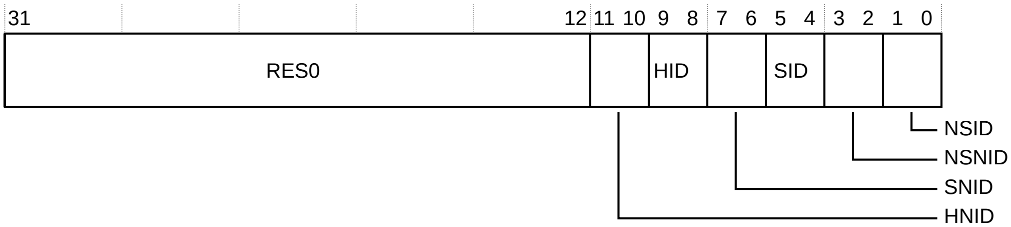
The AUTHSTATUS register characteristics are:

Attributes

Offset	`0x0FB8`
Type	Read-only
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

Figure 6-70: AUTHSTATUS register bit assignments



The following table shows the bit assignments.

Table 6-73: AUTHSTATUS register bit assignments

Bits	Reset value	Name	Function
[31:12]	`0x0`	RES0	Reserved bit or field with SBZP behavior.

Bits	Reset value	Name	Function
[11:10]	`0b00`	HNID	Hypervisor non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[9:8]	`0b00`	HID	Hypervisor invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[7:6]	UNKNOWN	SNID	Secure non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[5:4]	UNKNOWN	SID	Secure invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[3:2]	UNKNOWN	NSNID	Non-secure non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[1:0]	UNKNOWN	NSID	Non-secure invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled

6.4.2.21 Device Architecture Register, DEVARCH

Identifies the architect and architecture of a CoreSight component. The architect might differ from the designer of a component, for example Arm defines the architecture but another company designs and implements the component.

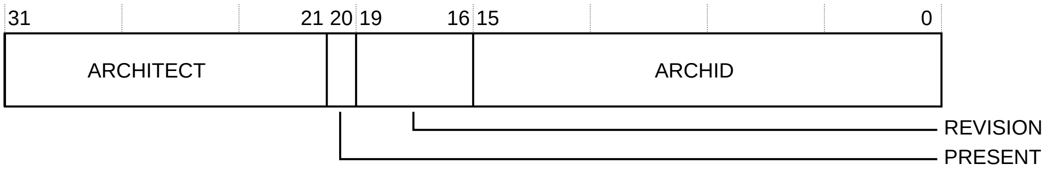
The DEVARCH register characteristics are:

Attributes

Offset	`0x0FBC`
Type	Read-only
Reset	`0x47700A17`
Width	32

The following figure shows the bit assignments.

Figure 6-71: DEVARCH register bit assignments



The following table shows the bit assignments.

Table 6-74: DEVARCH register bit assignments

Bits	Reset value	Name	Function
[31:21]	`0b01000111011`	ARCHITECT	Returns `0x23B`, denoting Arm as architect of the component
[20]	`0b1`	PRESENT	Returns 1, indicating that the DEVARCH register is present
[19:16]	`0b0000`	REVISION	Architecture revision. Returns the revision of the architecture that the ARCHID field specifies.
[15:0]	`0xA17`	ARCHID	Architecture ID. Returns `0x0A17`, identifying APv2 MEM-AP architecture v0.

6.4.2.22 Device Type Identifier Register, DEVTYPE

A debugger can use this register to get information about a component that has an unrecognized Part number.

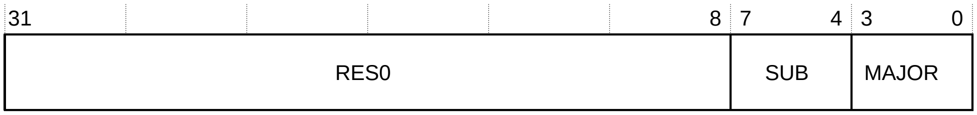
The DEVTYPE register characteristics are:

Attributes

Offset	`0x0FCC`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-72: DEVTYPE register bit assignments



The following table shows the bit assignments.

Table 6-75: DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:4]	`0b0000`	SUB	Minor classification. Returns `0x0`, Other/undefined.
[3:0]	`0b0000`	MAJOR	Major classification. Returns `0x0`, Miscellaneous.

6.4.2.23 Peripheral Identification Register 4, PIDR4

The PIDR4 register is part of the set of peripheral identification registers.

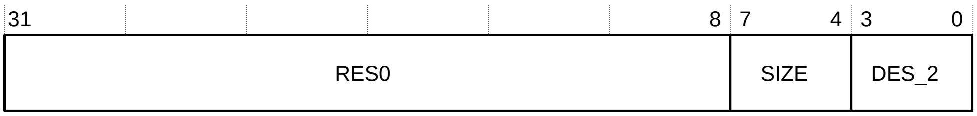
The PIDR4 register characteristics are:

Attributes

Offset	`0x0FD0`
Type	Read-only
Reset	`0x00000004`
Width	32

The following figure shows the bit assignments.

Figure 6-73: PIDR4 register bit assignments



The following table shows the bit assignments.

Table 6-76: PIDR4 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:4]	`0b0000`	SIZE	Indicates the memory size that is used by this component. Returns 0 indicating that the component uses an **UNKNOWN** number of 4KB blocks. Using the SIZE field to indicate the size of the component is deprecated.
[3:0]	`0b0100`	DES_2	JEP106 continuation code. Together, with PIDR2.DES_1 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same.

6.4.2.24 Peripheral Identification Register 5, PIDR5

The PIDR5 register is part of the set of peripheral identification registers.

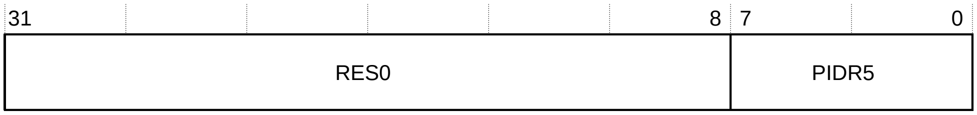
The PIDR5 register characteristics are:

Attributes

Offset	`0x0FD4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-74: PIDR5 register bit assignments



The following table shows the bit assignments.

Table 6-77: PIDR5 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:0]	`0b00000000`	PIDR5	Reserved

6.4.2.25 Peripheral Identification Register 6, PIDR6

The PIDR6 register is part of the set of peripheral identification registers.

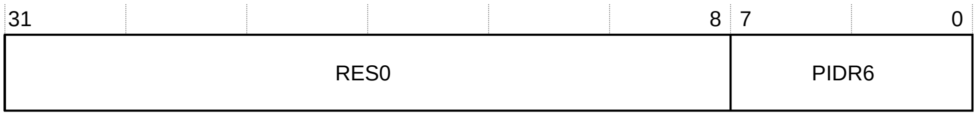
The PIDR6 register characteristics are:

Attributes

Offset	`0x0FD8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-75: PIDR6 register bit assignments



The following table shows the bit assignments.

Table 6-78: PIDR6 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:0]	`0b00000000`	PIDR6	Reserved

6.4.2.26 Peripheral Identification Register 7, PIDR7

The PIDR7 register is part of the set of peripheral identification registers.

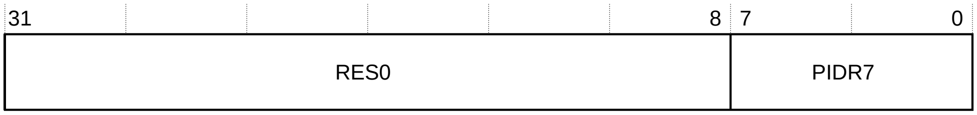
The PIDR7 register characteristics are:

Attributes

Offset	`0x0FDC`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-76: PIDR7 register bit assignments



The following table shows the bit assignments.

Table 6-79: PIDR7 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:0]	`0b00000000`	PIDR7	Reserved

6.4.2.27 Peripheral Identification Register 0, PIDR0

The PIDR0 register is part of the set of peripheral identification registers.

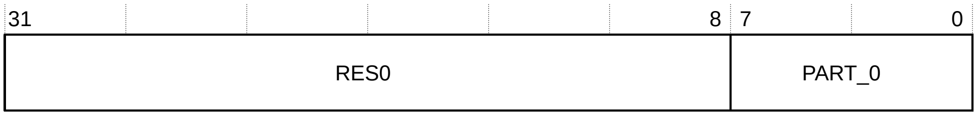
The PIDR0 register characteristics are:

Attributes

Offset	`0x0FE0`
Type	Read-only
Reset	`0x000000E3`
Width	32

The following figure shows the bit assignments.

Figure 6-77: PIDR0 register bit assignments



The following table shows the bit assignments.

Table 6-80: PIDR0 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior

Bits	Reset value	Name	Function
[7:0]	`0b11100011`	PART_0	Part number, bits[7:0]. Taken together with PIDR1.PART_1 it indicates the component. The Part Number is selected by the designer of the component.

6.4.2.28 Peripheral Identification Register 1, PIDR1

The PIDR1 register is part of the set of peripheral identification registers.

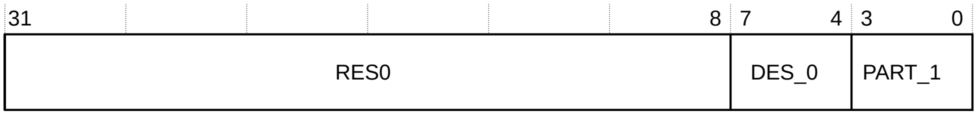
The PIDR1 register characteristics are:

Attributes

Offset	`0x0FE4`
Type	Read-only
Reset	`0x000000B9`
Width	32

The following figure shows the bit assignments.

Figure 6-78: PIDR1 register bit assignments



The following table shows the bit assignments.

Table 6-81: PIDR1 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:4]	`0b1011`	DES_0	JEP106 identification code, bits[3:0]. Together, with PIDR4.DES_2 and PIDR2.DES_1, they indicate the designer of the component and not the implementer, except where the two are the same.
[3:0]	`0b1001`	PART_1	Part number, bits[11:8]. Taken together with PIDR0.PART_0 it indicates the component. The Part Number is selected by the designer of the component.

6.4.2.29 Peripheral Identification Register 2, PIDR2

The PIDR2 register is part of the set of peripheral identification registers.

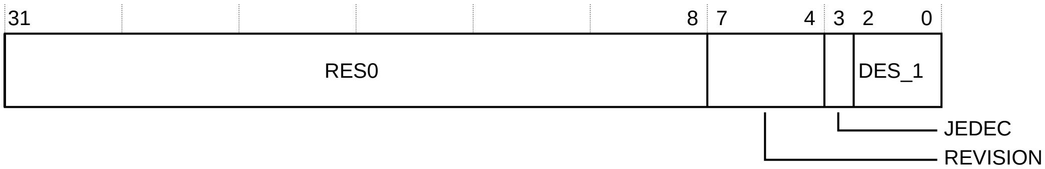
The PIDR2 register characteristics are:

Attributes

Offset	`0x0FE8`
Type	Read-only
Reset	`0x0000004B`
Width	32

The following figure shows the bit assignments.

Figure 6-79: PIDR2 register bit assignments



The following table shows the bit assignments.

Table 6-82: PIDR2 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:4]	`0b0100`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the Component list in Chapter 1 for information on the RTL revision of the component.
[3]	`0b1`	JEDEC	1 - Always set. Indicates that a JEDEC assigned value is used.
[2:0]	`0b011`	DES_1	JEP106 identification code, bits[6:4]. Together, with PIDR4.DES_2 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same.

6.4.2.30 Peripheral Identification Register 3, PIDR3

The PIDR3 register is part of the set of peripheral identification registers.

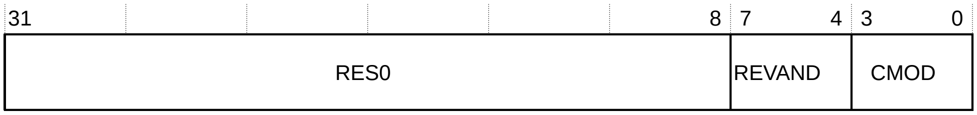
The PIDR3 register characteristics are:

Attributes

Offset	`0x0FEC`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-80: PIDR3 register bit assignments



The following table shows the bit assignments.

Table 6-83: PIDR3 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:4]	`0b0000`	REVAND	This field indicates minor errata fixes specific to this design, for example metal fixes after implementation. In most cases this field is `0x0`.
[3:0]	`0b0000`	CMOD	Customer Modified. Where the component is reusable IP, this value indicates if the customer has modified the behavior of the component. In most cases this field is `0x0`.

6.4.2.31 Component Identification Register 0, CIDR0

The CIDR0 register is part of the set of component identification registers.

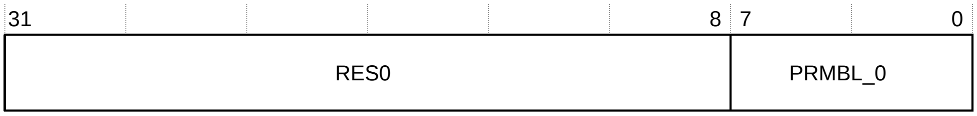
The CIDR0 register characteristics are:

Attributes

Offset	`0x0FF0`
Type	Read-only
Reset	`0x0000000D`
Width	32

The following figure shows the bit assignments.

Figure 6-81: CIDR0 register bit assignments



The following table shows the bit assignments.

Table 6-84: CIDR0 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

6.4.2.32 Component Identification Register 1, CIDR1

The CIDR1 register is part of the set of component identification registers.

The CIDR1 register characteristics are:

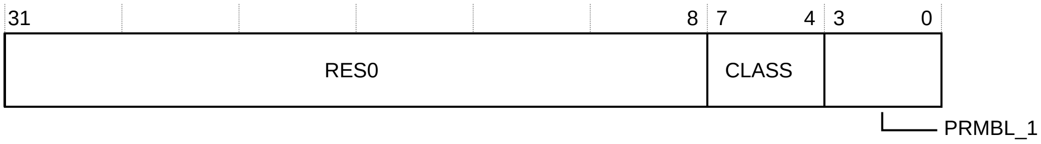
Attributes

Offset	`0x0FF4`
Type	Read-only
Reset	`0x00000090`

Width 32

The following figure shows the bit assignments.

Figure 6-82: CIDR1 register bit assignments



The following table shows the bit assignments.

Table 6-85: CIDR1 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

6.4.2.33 Component Identification Register 2, CIDR2

The CIDR2 register is part of the set of component identification registers.

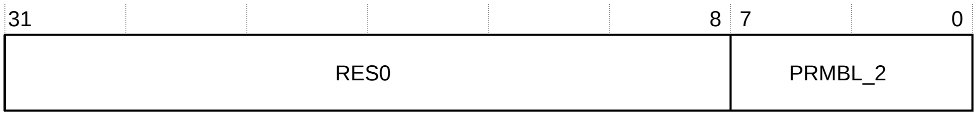
The CIDR2 register characteristics are:

Attributes

Offset `0x0FF8`
Type Read-only
Reset `0x00000005`
Width 32

The following figure shows the bit assignments.

Figure 6-83: CIDR2 register bit assignments



The following table shows the bit assignments.

Table 6-86: CIDR2 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

6.4.2.34 Component Identification Register 3, CIDR3

The CIDR3 register is part of the set of component identification registers.

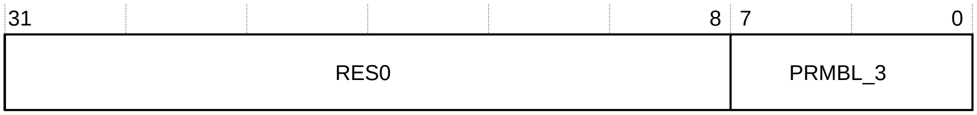
The CIDR3 register characteristics are:

Attributes

Offset	`0x0FFC`
Type	Read-only
Reset	`0x000000B1`
Width	32

The following figure shows the bit assignments.

Figure 6-84: CIDR3 register bit assignments



The following table shows the bit assignments.

Table 6-87: CIDR3 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with SBZP behavior
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

6.5 css600_apbrom introduction

This section describes the programmers model of the `css600_apbrom`.

6.5.1 `css600_apbrom` register summary

The following table shows the registers in offset order from the base memory address.

Note

A reset value containing one or more '-' means that this register contains **UNKNOWN** or **IMPLEMENTATION-DEFINED** values. See the relevant register description for more information.

Locations that are not listed in the table are Reserved.

Table 6-88: `css600_apbrom - APB4_Slave_0` register summary

Offset	Name	Type	Reset	Width	Description
0x0000	ROMEntry0	RO	0x--------	32	6.5.2.1 ROM Entries register 0, ROMEntry0 on page 105
0x0FB8	AUTHSTATUS	RO	0x000000--	32	6.5.2.5 Authentication Status Register, AUTHSTATUS on page 109
0x0FBC	DEVARCH	RO	0x47700AF7	32	6.5.2.6 Device Architecture Register, DEVARCH on page 110
0x0FC8	DEVID	RO	0x000000-0	32	6.5.2.7 Device Configuration Register, DEVID on page 111
0x0FD0	PIDR4	RO	0x0000000-	32	6.5.2.8 Peripheral Identification Register 4, PIDR4 on page 112
0x0FD4	PIDR5	RO	0x00000000	32	6.5.2.9 Peripheral Identification Register 5, PIDR5 on page 113
0x0FD8	PIDR6	RO	0x00000000	32	6.5.2.10 Peripheral Identification Register 6, PIDR6 on page 113
0x0FDC	PIDR7	RO	0x00000000	32	6.5.2.11 Peripheral Identification Register 7, PIDR7 on page 114
0x0FE0	PIDR0	RO	0x000000--	32	6.5.2.12 Peripheral Identification Register 0, PIDR0 on page 114
0x0FE4	PIDR1	RO	0x000000--	32	6.5.2.13 Peripheral Identification Register 1, PIDR1 on page 115
0x0FE8	PIDR2	RO	0x000000--	32	6.5.2.14 Peripheral Identification Register 2, PIDR2 on page 116
0x0FEC	PIDR3	RO	0x00000000	32	6.5.2.15 Peripheral Identification Register 3, PIDR3 on page 116
0x0FF0	CIDR0	RO	0x0000000D	32	6.5.2.16 Component Identification Register 0, CIDR0 on page 117
0x0FF4	CIDR1	RO	0x00000090	32	6.5.2.17 Component Identification Register 1, CIDR1 on page 118
0x0FF8	CIDR2	RO	0x00000005	32	6.5.2.18 Component Identification Register 2, CIDR2 on page 118
0x0FFC	CIDR3	RO	0x000000B1	32	6.5.2.19 Component Identification Register 3, CIDR3 on page 119

6.5.2 `css600_apbrom` register descriptions

This section describes the `css600_apbrom` registers.

6.5.1 css600_apbrom register summary on page 104 provides cross references to individual registers.

6.5.2.1 ROM Entries register 0, ROMEntry0

Each register contains a descripter of a CoreSight component in the system. All ROM table entries conform to the same format.

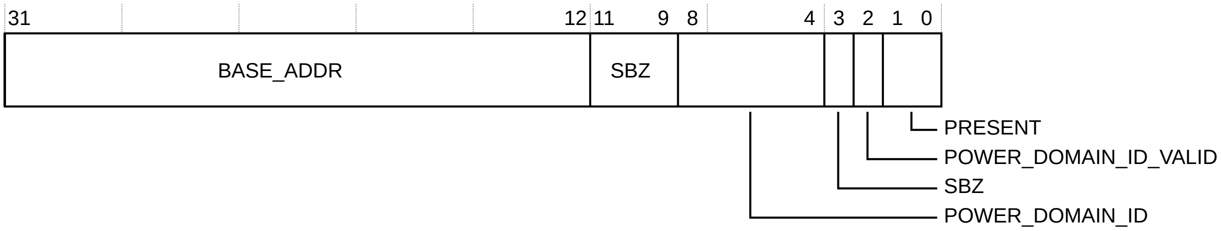
The ROMEntry0 register characteristics are:

Attributes

Offset	0x0000
Type	Read-only
Reset	0x--------
Width	32

The following figure shows the bit assignments.

Figure 6-85: ROMEntry0 register bit assignments



The following table shows the bit assignments.

Table 6-89: ROMEntry0 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION DEFINED	BASE_ADDR	Base address of component
[11:9]	0b000	SBZ	Software should write the field as all 0s
[8:4]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID	Indicates the power domain ID of the component. Only valid if bit 2 is set. If bit 2 is clear then this field has a value of 0. Possible values are 0 to 31, representing the 32 DBGPWRUPREQ/ACK interface pins of the component.
[3]	0b0	SBZ	Software should write the field as all 0s
[2]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID_VALID	Indicates whether there is a power domain ID specified in the ROM table entry: **0** POWER_DOMAIN_ID field of this register is not valid **1** POWER_DOMAIN_ID field of this register is valid
[1:0]	IMPLEMENTATION DEFINED	PRESENT	Indicates whether the ROM table entry is present: **0x0** ROM table entry not present. This is the last entry. **0x1** Reserved **0x2** ROM table entry not present. This is not the last entry. **0x3** ROM table entry present

6.5.2.2 ROM Entries register 1, ROMEntry1

Each register contains a descripter of a CoreSight component in the system. All ROM table entries conform to the same format.

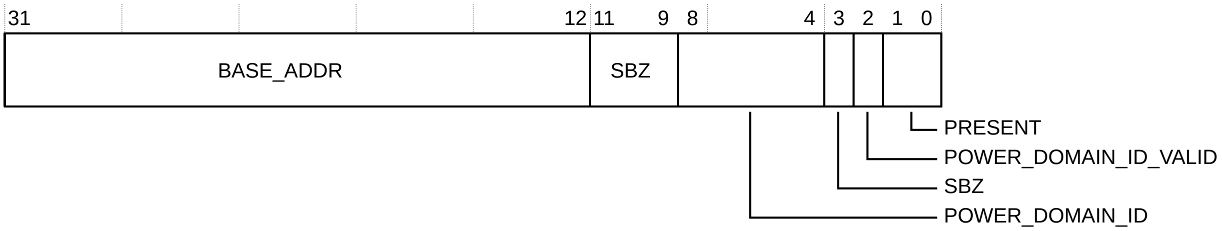
The ROMEntry1 register characteristics are:

Attributes

Offset	0x0004
Type	Read-only
Reset	0x--------
Width	32

The following figure shows the bit assignments.

Figure 6-86: ROMEntry1 register bit assignments



The following table shows the bit assignments.

Table 6-90: ROMEntry1 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION DEFINED	BASE_ADDR	Base address of component
[11:9]	`0b000`	SBZ	Software should write the field as all 0s
[8:4]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID	Indicates the power domain ID of the component. Only valid if bit 2 is set. If bit 2 is clear then this field has a value of 0. Possible values are 0 to 31, representing the 32 DBGPWRUPREQ/ACK interface pins of the component.
[3]	`0b0`	SBZ	Software should write the field as all 0s
[2]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID_VALID	Indicates whether there is a power domain ID specified in the ROM table entry: **0** POWER_DOMAIN_ID field of this register is not valid **1** POWER_DOMAIN_ID field of this register is valid
[1:0]	IMPLEMENTATION DEFINED	PRESENT	Indicates whether the ROM table entry is present: **0x0** ROM table entry not present. This is the last entry. **0x1** Reserved **0x2** ROM table entry not present. This is not the last entry. **0x3** ROM table entry present

6.5.2.3 ROM Entries register 2, ROMEntry2

Each register contains a descripter of a CoreSight component in the system. All ROM table entries conform to the same format.

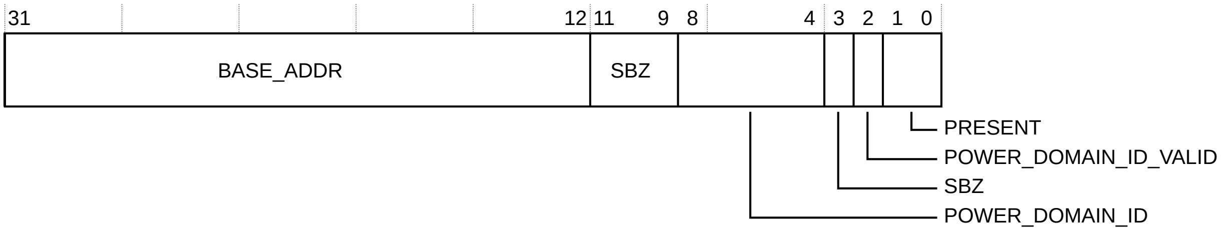
The ROMEntry2 register characteristics are:

Attributes

Offset	`0x0008`
Type	Read-only
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

Figure 6-87: ROMEntry2 register bit assignments



The following table shows the bit assignments.

Table 6-91: ROMEntry2 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION DEFINED	BASE_ADDR	Base address of component
[11:9]	0b000	SBZ	Software should write the field as all 0s
[8:4]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID	Indicates the power domain ID of the component. Only valid if bit 2 is set. If bit 2 is clear then this field has a value of 0. Possible values are 0 to 31, representing the 32 DBGPWRUPREQ/ACK interface pins of the component.
[3]	0b0	SBZ	Software should write the field as all 0s
[2]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID_VALID	Indicates whether there is a power domain ID specified in the ROM table entry: **0** POWER_DOMAIN_ID field of this register is not valid **1** POWER_DOMAIN_ID field of this register is valid
[1:0]	IMPLEMENTATION DEFINED	PRESENT	Indicates whether the ROM table entry is present **0x0** ROM table entry not present. This is the last entry. **0x1** Reserved **0x2** ROM table entry not present. This is not the last entry. **0x3** ROM table entry present

6.5.2.4 ROM Entries register 511, ROMEntry511

Each register contains a descripter of a CoreSight component in the system. All ROM table entries conform to the same format.

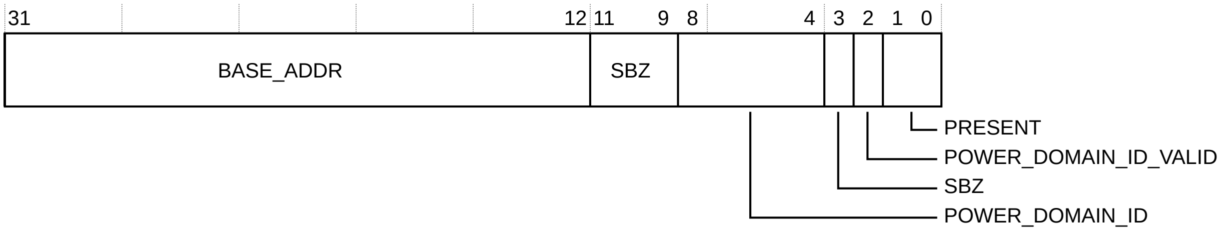
The ROMEntry511 register characteristics are:

Attributes

Offset	0x07FC
Type	Read-only
Reset	0x--------
Width	32

The following figure shows the bit assignments.

Figure 6-88: ROMEntry511 register bit assignments



The following table shows the bit assignments.

Table 6-92: ROMEntry511 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION DEFINED	BASE_ADDR	Base address of component
[11:9]	`0b000`	SBZ	Software should write the field as all 0s
[8:4]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID	Indicates the power domain ID of the component. Only valid if bit 2 is set. If bit 2 is clear then this field has a value of 0. Possible values are 0 to 31, representing the 32 DBGPWRUPREQ/ACK interface pins of the component.
[3]	`0b0`	SBZ	Software should write the field as all 0s
[2]	IMPLEMENTATION DEFINED	POWER_DOMAIN_ID_VALID	Indicates whether there is a power domain ID specified in the ROM table entry: **0** POWER_DOMAIN_ID field of this register is not valid **1** POWER_DOMAIN_ID field of this register is valid
[1:0]	IMPLEMENTATION DEFINED	PRESENT	Indicates whether the ROM table entry is present: **`0x0`** ROM table entry not present. This is the last entry. **`0x1`** Reserved **`0x2`** ROM table entry not present. This is not the last entry. **`0x3`** ROM table entry present

6.5.2.5 Authentication Status Register, AUTHSTATUS

Reports the current status of the authentication control signals.

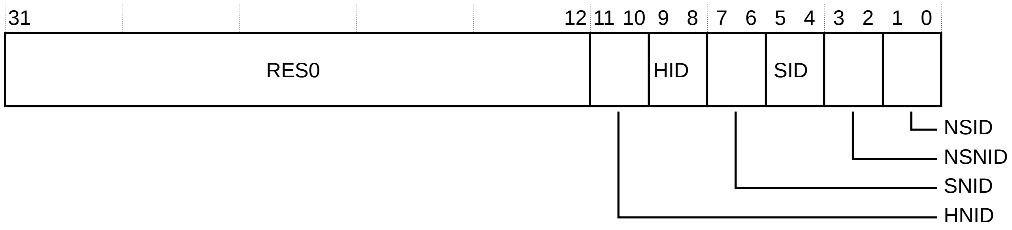
The AUTHSTATUS register characteristics are:

Attributes

Offset	`0x0FB8`
Type	Read-only
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

Figure 6-89: AUTHSTATUS register bit assignments



The following table shows the bit assignments.

Table 6-93: AUTHSTATUS register bit assignments

Bits	Reset value	Name	Function
[31:12]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[11:10]	`0b00`	HNID	Hypervisor non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[9:8]	`0b00`	HID	Hypervisor invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[7:6]	UNKNOWN	SNID	Secure non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[5:4]	UNKNOWN	SID	Secure invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[3:2]	UNKNOWN	NSNID	Non-secure non-invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled
[1:0]	UNKNOWN	NSID	Non-secure invasive debug: **0x0** Functionality not implemented or controlled elsewhere **0x1** Reserved **0x2** Functionality disabled **0x3** Functionality enabled

6.5.2.6 Device Architecture Register, DEVARCH

Identifies the architect and architecture of a CoreSight component. The architect might differ from the designer of a component, for example Arm defines the architecture but another company designs and implements the component.

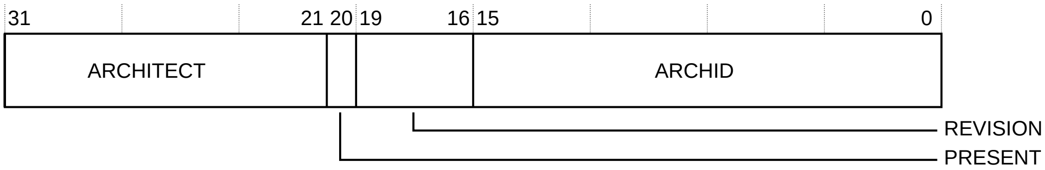
The DEVARCH register characteristics are:

Attributes

Offset	`0x0FBC`
Type	Read-only
Reset	`0x47700AF7`
Width	32

The following figure shows the bit assignments.

Figure 6-90: DEVARCH register bit assignments



The following table shows the bit assignments.

Table 6-94: DEVARCH register bit assignments

Bits	Reset value	Name	Function
[31:21]	`0b01000111011`	ARCHITECT	Returns `0x23B`, denoting Arm as architect of the component
[20]	`0b1`	PRESENT	Returns 1, indicating that the DEVARCH register is present
[19:16]	`0b0000`	REVISION	Architecture revision. Returns the revision of the architecture that the ARCHID field specifies.
[15:0]	`0xAF7`	ARCHID	Architecture ID. Returns `0x0AF7`, identifying ROM Table Architecture v0.

6.5.2.7 Device Configuration Register, DEVID

This register is IMPLEMENTATION DEFINED for each Part Number and Designer. The register indicates the capabilities of the component.

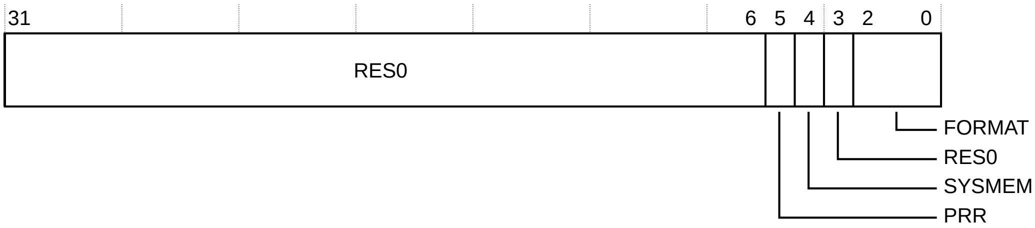
The DEVID register characteristics are:

Attributes

Offset	`0x0FC8`
Type	Read-only
Reset	`0x000000-0`
Width	32

The following figure shows the bit assignments.

Figure 6-91: DEVID register bit assignments



The following table shows the bit assignments.

Table 6-95: DEVID register bit assignments

Bits	Reset value	Name	Function
[31:6]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[5]	`0b0`	PRR	Indicates that power request functionality is included. Set by the GPR_PRESENT parameter: **0** GPR is not included (css600_apbrom) **1** GPR is included (css600_apbrom_gpr)
[4]	IMPLEMENTATION DEFINED	*SYSMEM*	Indicates whether system memory is present on the bus. Set by the *SYSMEM* parameter: **0** System memory is not present and the bus is a dedicated debug bus **1** Indicates that there is system memory on the bus
[3]	`0b0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[2:0]	`0b000`	FORMAT	Indicates that this is a 32-bit ROM Table

6.5.2.8 Peripheral Identification Register 4, PIDR4

The PIDR4 register is part of the set of peripheral identification registers.

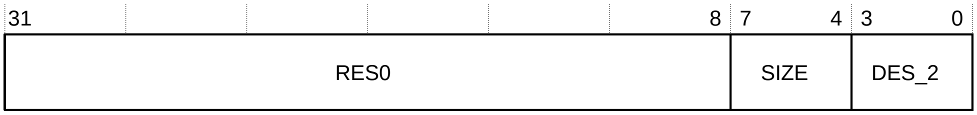
The PIDR4 register characteristics are:

Attributes

Offset	`0x0FD0`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

Figure 6-92: PIDR4 register bit assignments



The following table shows the bit assignments.

Table 6-96: PIDR4 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b0000`	SIZE	Indicates the memory size that is used by this component. Returns 0 indicating that the component uses an UNKNOWN number of 4KB blocks. Using the SIZE field to indicate the size of the component is deprecated.
[3:0]	IMPLEMENTATION DEFINED	DES_2	JEP106 continuation code. Together, with PIDR2.DES_1 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same.

6.5.2.9 Peripheral Identification Register 5, PIDR5

The PIDR5 register is part of the set of peripheral identification registers.

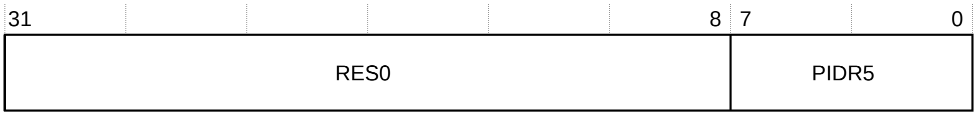
The PIDR5 register characteristics are:

Attributes

Offset	`0x0FD4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-93: PIDR5 register bit assignments



The following table shows the bit assignments.

Table 6-97: PIDR5 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000000`	PIDR5	Reserved

6.5.2.10 Peripheral Identification Register 6, PIDR6

The PIDR6 register is part of the set of peripheral identification registers.

The PIDR6 register characteristics are:

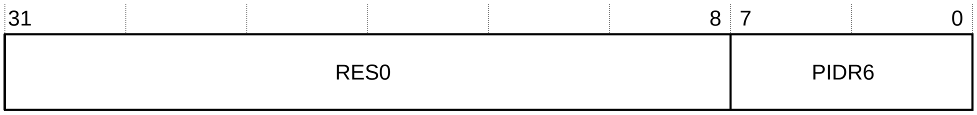
Attributes

Offset	`0x0FD8`
Type	Read-only
Reset	`0x00000000`

Width 32

The following figure shows the bit assignments.

Figure 6-94: PIDR6 register bit assignments



The following table shows the bit assignments.

Table 6-98: PIDR6 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000000`	PIDR6	Reserved

6.5.2.11 Peripheral Identification Register 7, PIDR7

The PIDR7 register is part of the set of peripheral identification registers.

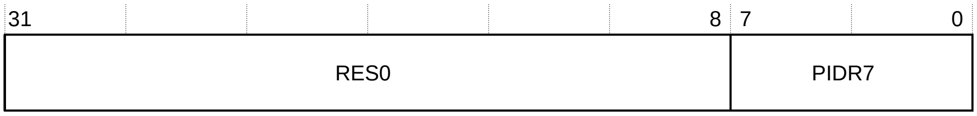
The PIDR7 register characteristics are:

Attributes

Offset `0x0FDC`
Type Read-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

Figure 6-95: PIDR7 register bit assignments



The following table shows the bit assignments.

Table 6-99: PIDR7 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000000`	PIDR7	Reserved

6.5.2.12 Peripheral Identification Register 0, PIDR0

The PIDR0 register is part of the set of peripheral identification registers.

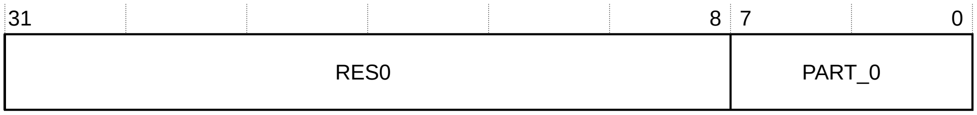
The PIDR0 register characteristics are:

Attributes

Offset	`0x0FE0`
Type	Read-only
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

Figure 6-96: PIDR0 register bit assignments



The following table shows the bit assignments.

Table 6-100: PIDR0 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	IMPLEMENTATION DEFINED	PART_0	Part number, bits[7:0]. Set by the configuration inputs **part_number[7:0]**

6.5.2.13 Peripheral Identification Register 1, PIDR1

The PIDR1 register is part of the set of peripheral identification registers.

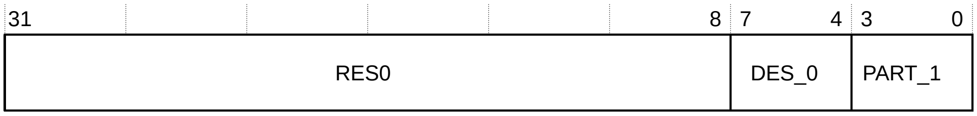
The PIDR1 register characteristics are:

Attributes

Offset	`0x0FE4`
Type	Read-only
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

Figure 6-97: PIDR1 register bit assignments



The following table shows the bit assignments.

Table 6-101: PIDR1 register bit assignments

Bits	Reset value	Name	Function
[31:8]	0x0	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	IMPLEMENTATION DEFINED	DES_0	JEP106 identification code, bits[3:0]. Set by the configuration inputs **jep106_id[3:0]**. Together, with PIDR4.DES_2 and PIDR2.DES_1, they indicate the designer of the component and not the implementer, except where the two are the same.
[3:0]	IMPLEMENTATION DEFINED	PART_1	Part number, bits[11:8]. Set by the configuration inputs **part_number[11:8]**.

6.5.2.14 Peripheral Identification Register 2, PIDR2

The PIDR2 register is part of the set of peripheral identification registers.

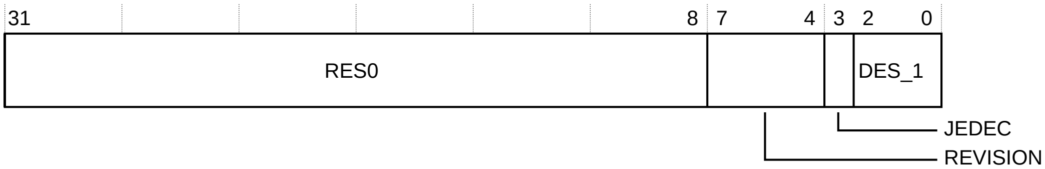
The PIDR2 register characteristics are:

Attributes

Offset	0x0FE8
Type	Read-only
Reset	0x000000--
Width	32

The following figure shows the bit assignments.

Figure 6-98: PIDR2 register bit assignments



The following table shows the bit assignments.

Table 6-102: PIDR2 register bit assignments

Bits	Reset value	Name	Function
[31:8]	0x0	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	IMPLEMENTATION DEFINED	REVISION	Revision. Set by the configuration inputs **revision[3:0]**.
[3]	0b1	JEDEC	1 - Always set. Indicates that a JEDEC assigned value is used.
[2:0]	IMPLEMENTATION DEFINED	DES_1	JEP106 identification code, bits[6:4]. Set by the configuration inputs **jep106_id[6:4]**. Together, with PIDR4.DES_2 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same.

6.5.2.15 Peripheral Identification Register 3, PIDR3

The PIDR3 register is part of the set of peripheral identification registers.

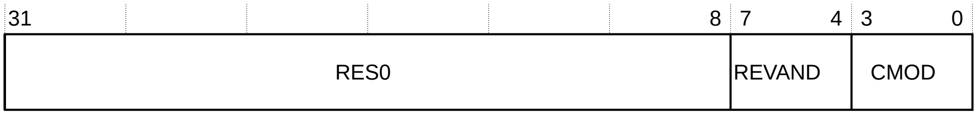
The PIDR3 register characteristics are:

Attributes

Offset	`0x0FEC`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 6-99: PIDR3 register bit assignments



The following table shows the bit assignments.

Table 6-103: PIDR3 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b0000`	REVAND	This field indicates minor errata fixes specific to this design, for example metal fixes after implementation. In most cases this field is `0x0`.
[3:0]	`0b0000`	CMOD	Customer Modified. Where the component is reusable IP, this value indicates if the customer has modified the behavior of the component. In most cases this field is `0x0`.

6.5.2.16 Component Identification Register 0, CIDR0

The CIDR0 register is part of the set of component identification registers.

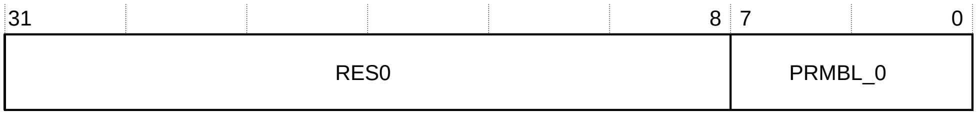
The CIDR0 register characteristics are:

Attributes

Offset	`0x0FF0`
Type	Read-only
Reset	`0x0000000D`
Width	32

The following figure shows the bit assignments.

Figure 6-100: CIDR0 register bit assignments



The following table shows the bit assignments.

Table 6-104: CIDR0 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

6.5.2.17 Component Identification Register 1, CIDR1

The CIDR1 register is part of the set of component identification registers.

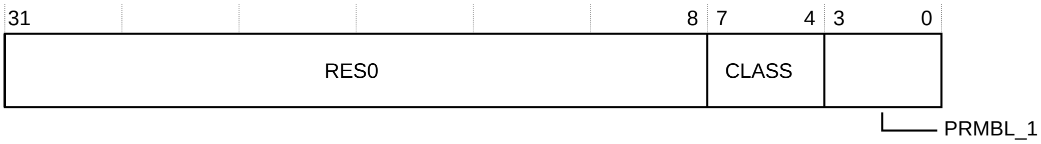
The CIDR1 register characteristics are:

Attributes

Offset	`0x0FF4`
Type	Read-only
Reset	`0x00000090`
Width	32

The following figure shows the bit assignments.

Figure 6-101: CIDR1 register bit assignments



The following table shows the bit assignments.

Table 6-105: CIDR1 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

6.5.2.18 Component Identification Register 2, CIDR2

The CIDR2 register is part of the set of component identification registers.

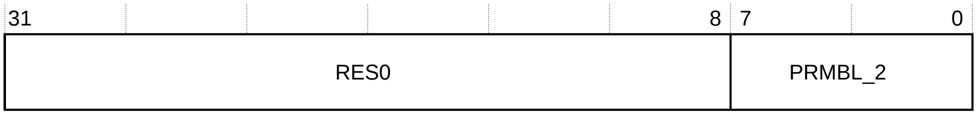
The CIDR2 register characteristics are:

Attributes

Offset	`0x0FF8`
Type	Read-only
Reset	`0x00000005`
Width	32

The following figure shows the bit assignments.

Figure 6-102: CIDR2 register bit assignments



The following table shows the bit assignments.

Table 6-106: CIDR2 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

6.5.2.19 Component Identification Register 3, CIDR3

The CIDR3 register is part of the set of component identification registers.

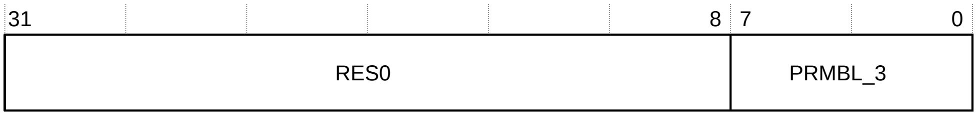
The CIDR3 register characteristics are:

Attributes

Offset	`0x0FFC`
Type	Read-only
Reset	`0x000000B1`
Width	32

The following figure shows the bit assignments.

Figure 6-103: CIDR3 register bit assignments



The following table shows the bit assignments.

Table 6-107: CIDR3 register bit assignments

Bits	Reset value	Name	Function
[31:8]	`0x0`	RES0	Reserved bit or field with Should-Be-Zero-or-Preserved (SBZP) behavior
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

Appendix A Revisions

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

A.1 Revisions

Each table shows the technical differences between successive issues of the document.

Table A-1: Issue 0000-00

Change	Location
First release	-

Table A-2: Differences between issue 0100-00 and issue 0000-00

Change	Location
DAP for Arm Cortex-M processor added	Introductory information in: 3.2 DAP-Lite2 for Arm Cortex-M processors on page 17 5.2 DAP-Lite2 for Arm Cortex-M processors on page 32
Both DAPs built from SoC-600 components	4 SoC-600 components functional description on page 20 6 SoC-600 components programmers model on page 35
Support for CoreSight Architecture v3.0	See *Arm® CoreSight™ Architecture Specification v3.0*
Support for Arm Debug Interface v6.0	See *Arm® Debug Interface Architecture Specification ADIv6.0*
JTAG DP supports configurable IR length	See *Arm® CoreSight™ DAP-Lite2 Configuration and Integration Manual r1p0*
SW DP Serial Wire Multi Drop support added	See *Arm® CoreSight™ DAP-Lite2 Configuration and Integration Manual r1p0*
Q-Channel support for low power implementations	See *Arm® CoreSight™ DAP-Lite2 Configuration and Integration Manual r1p0*
Programmers model details expanded	5 DAP-Lite2 programmers model on page 30

Table A-3: Differences between issue 0100-00 and issue 0101-00

Change	Location
Updated DP components	2.6 Product revisions on page 15, 6.2.1 css600_dp register summary on page 36, and 6.2.2.3 css600_dp Debug Port Identification Register, DPIDR on page 39.

Table A-4: Differences between issue 0101-00 and issue 0200-00

Change	Location
Updated AHB-AP component	2.6 Product revisions on page 15, 6.4.2.5 Control Status Word register, CSW on page 82, 6.4.2.15 Identification Register, IDR on page 90, and 6.4.2.29 Peripheral Identification Register 2, PIDR2 on page 100.

Table A-5: Differences between issue 0200-00 and issue 0201-00

Change	Location
Updated publishing style, including section numbering, and edits	Throughout the document

Change	Location
Updated APB-AP and AHB-AP components, IDR and PIDR2 registers	• 2.5 Component list on page 14 • 6.3.1 css600_apbap register summary on page 50 • 6.3.2.15 Identification Register, IDR on page 63 • 6.3.2.29 Peripheral Identification Register 2, PIDR2 on page 73 • 6.4.1 css600_ahbap - APB4_Slave_0 register summary on page 77 • 6.4.2.15 Identification Register, IDR on page 90 • 6.4.2.29 Peripheral Identification Register 2, PIDR2 on page 100
Updated APB CSW.prot register field	6.3.2.5 Control Status Word register, CSW on page 55
Updated AHB CSW.HNONSEC register field	#unique_174