ARM® CoreSight™ SoC-600

Revision: r1p0

Technical Reference Manual

ARM

ARM 100806_0100_00_en

ARM® CoreSight™ SoC-600

Technical Reference Manual

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	01 March 2017	Non-Confidential	First release for r0p0
0000-01	11 May 2017	Non-Confidential	Second release for r0p0
0100-00	01 August 2017	Non-Confidential	First release for r1p0

Non-Confidential Proprietary Notice

ARM Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

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

Unrestricted Access is an ARM internal classification.

Product Status

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

Web Address

http://www.arm.com

Contents
ARM® CoreSight™ SoC-600 Technical Reference Manual

Preface

This preface introduces the *ARM® CoreSight™ SoC-600 Technical Reference Manual*.

It contains the following:

About this book

ARM CoreSight SoC-600 Technical Reference Manual. This book describes the CoreSight SoC-600 System Components and the features available in them.

Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this book, for example, r*1*p*2*, where:

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

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

Intended audience

This book is written for the following audiences:

- Hardware and software engineers who want to incorporate CoreSight™ SoC-600 into their design and produce real-time instruction and data trace information from a SoC.
- Software engineers writing tools to use CoreSight SoC-600.

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

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction
This chapter introduces the CoreSight SoC-600.

Chapter 2 DAP components functional description
This chapter describes the functionality of the SoC-600.

Chapter 3 APB infrastructure components functional description
This chapter describes the functionality of the APB infrastructure components.

Chapter 4 ATB infrastructure components functional description
This chapter describes the functionality of the ATB infrastructure components.

Chapter 5 Timestamp components functional description
This chapter describes the functionality of the timestamp components.

Chapter 6 Embedded Cross Trigger components functional description
The cross-triggering components enable CoreSight components to broadcast events between each other.

Chapter 7 Authentication components functional description
This chapter describes the functionality of the authentication components.

Chapter 8 Processor Integration Layer components
This chapter gives an overview of the Cortex-M4 *Processor Integration Layer* (PIL).

Chapter 9 Programmers model
This chapter describes the programmers models for all CoreSight SoC-600 components that have programmable registers.

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

Glossary

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

See the *ARM® Glossary* for more information.

Typographic conventions

italic
: Introduces special terminology, denotes cross-references, and citations.

bold
: Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

`monospace`
: Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

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

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

`monospace bold`
: Denotes language keywords when used outside example code.

`<and>`
: Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example:

```
MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>
```

SMALL CAPITALS
: Used in body text for a few terms that have specific technical meanings, that are defined in the *ARM® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Timing diagrams

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

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

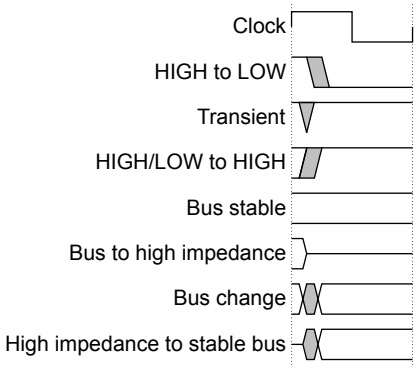

Figure 1 Key to timing diagram conventions

Signals

The signal conventions are:

Signal level

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

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

Lowercase n

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

Additional reading

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

ARM publications

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

- *ARM® CoreSight™ Architecture Specification v3.0* (ARM IHI 0029).
- *ARM® AMBA® APB Protocol Specification Version 2.0* (ARM IHI 0024).
- *ARM® AMBA® AXI and ACE Protocol Specification* (ARM IHI 0022).
- *ARM® AMBA® 4 ATB Protocol Specification* (ARM IHI 0032).
- *ARM® AMBA® 5 AHB Protocol Specification AHB5, AHB-Lite* (ARM IHI 0033).
- *ARM® Debug Interface Architecture Specification ADIv6.0* (ARM IHI 0074).
- *ARM® CoreLink™ LPD-500 Low Power Distributor Technical Reference Manual* (ARM 100361).
- *AMBA® Low Power Interface Specification, ARM® Q-Channel and P-Channel Interfaces* (ARM IHI 0068).
- *ARM® Architecture Reference Manual ARMv7-A and ARMv7-R edition* (ARM DDI 0406).
- *ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile* (ARM DDI 0487).

The following confidential books are only available to licensees:

- *ARM® CoreSight™ SoC-600 Configuration and Integration Manual* (ARM 100807).
- *ARM® Socrates™ System Builder User Guide* (ARM 100328).
- *ARM® Socrates™ System Builder Installation Guide* (ARM 100329).
- *ARM® Cortex®-M4 Integration and Implementation Manual* (ARM DII 0239).
- *ARM® Power Control System Architecture Specification Version 1.0* (ARM DEN 0050).

Other publications

- *Verilog-2001 Standard* (IEEE Std 1364-2001).

Feedback

Feedback on this product

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

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

Feedback on content

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

- The title *ARM CoreSight SoC-600 Technical Reference Manual*.
- The number ARM 100806_0100_00_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

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

Chapter 1
Introduction

This chapter introduces the CoreSight SoC-600.

It contains the following sections:

1.1 About CoreSight SoC-600

CoreSight SoC-600 is a member of the ARM embedded debug and trace component family.

Some of the features that CoreSight SoC-600 provides are:

- It is a collection of components that can be used for debug and trace of ARM SoCs, from simple single-processor designs through to complex multiprocessor and multi-cluster designs that include many heterogeneous processors.
- It supports the *ARM® Debug Interface* (ADI) v6 and CoreSight v3 Architectures that enable you to build debug and trace functionality into your systems, and to support debug and trace over existing functional interfaces.
- It is designed for simple integration using ARM Socrates System Builder for integration with other IP.
- CoreSight SoC-600 components support the development of low-power system implementations through architected fine-grained power control, and Q-Channel interfaces for clock and power quiescence.
- The ARM CoreLink LPD-500 can be integrated with CoreSight SoC-600 as part of a full-chip power and clock control methodology.

The CoreSight SoC-600 bundle includes:

- A library of configurable CoreSight components that are written in Verilog, and compliant with the *Verilog-2001 Standard* (IEEE Std 1364-2001).
- An ARM Socrates IP Tooling 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.
- Example timing constraint files for each component in SDC format.
- Constraint UPF files at component level for signals that are able to cross power domain boundaries.

1.2 Features

Features and capabilities that are provided by SoC-600 include:

Debug

- *ARM® Debug Interface Architecture Specification ADIv6.0*-compliant debug port. This debug port supports JTAG and Serial Wire protocols for connection to an off-chip debugger. This connection is achieved using a low-pin-count connection that is suitable for bare-metal debug and silicon bring-up.
- *ARM® CoreSight™ Architecture Specification v3.0* compliance enables debug over functional interfaces, suitable for application development and in-field debug without a dedicated debug interface.
- Infrastructure components supporting system identification and integration with other CoreSight IP.

Trace

- Versatile *Trace Memory Controller* (TMC) supporting local on-chip storage, and buffering of trace data.
- TMC router configuration supports efficient hand-off of trace data to other system masters. This feature enables trace over functional interfaces, suitable for application development and in-field debug without a dedicated debug and trace interface.
- TMC streaming configuration supports integration to third-party *High Speed Serial Trace Port*s (HSSTP) for high bandwidth, low pin count trace solutions.
- Infrastructure components supporting filtering and routing of trace data on chip.

Embedded Cross Triggering

- *Cross Trigger Interface* (CTI) supports up to 32 trigger inputs and outputs with a single component instance.
- *Cross Trigger Matrix* (CTM) supports up to 33 CTI or CTM connections without cascading.

Power

- *ARM® CoreSight™ Architecture Specification v3.0*-compliant *Granular Power Requester* (GPR) enables fine-grained debug and system power control at all levels of debug hierarchy.
- Components are designed for low-power implementation, supporting clock and power quiescence and wakeup signaling where necessary.
- Components support Q-Channel *Low-Power Interfaces* (LPI) for integration with power controllers to support system-level clock and power gating where necessary.
- Infrastructure components support implementation across multiple clock and power domains.

Miscellaneous

- Some components, such as the bridges and *Serial Wire Debug Port* (SW-DP), use two Verilog modules to span clock and power domains. This design can ease implementation in complex SoC designs that have multiple clock and power domains.
- Infrastructure components support integration with legacy IP including *ARM® CoreSight™ Architecture Specification v2.0*-compliant, and JTAG components.

1.3 Supported standards

CoreSight SoC-600 is compliant with several standards.

The standards are:

- *ARM® CoreSight™ Architecture Specification v3.0*.
- *ARM® AMBA® APB Protocol Specification Version 2.0*.
- *ARM® AMBA® 4 ATB Protocol Specification ATBv1.0 and ATBv1.1*.
- *ARM® Debug Interface Architecture Specification ADIv6.0*.
- *ARM® AMBA® 5 AHB Protocol Specification AHB5, AHB-Lite*.
- *ARM® AMBA® 4 AXI4-Stream Protocol Specification*.
- Accellera, *IP-XACT version 1685-2009*.
- *IEEE 1149.1-2001 IEEE Standard Test Access Port and Boundary Scan Architecture (JTAG)*.

1.4 Documentation

The SoC-600 documentation includes a *Technical Reference Manual* (TRM) and a *Configuration and Integration Manual* (CIM). These books relate to the SoC-600 design flow.

Technical Reference Manual

The TRM describes the functionality and the effects of functional options on the behavior of the SoC-600 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 SoC-600 components, then contact the integrator to determine the configuration of the device that you are using.

Configuration and Integration Manual

The CIM describes:

- How to configure the SoC-600 components.
- How to integrate the SoC-600 components into your SoC design and how to configure system-specific Identification Registers.
- How to implement the SoC-600 components to produce a hard macrocell of the design. This description includes custom cell replacement, describes the power domains, and design synthesis.

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

1.5 Design process

The SoC-600 components are delivered as synthesizable Verilog RTL.

Before the SoC-600 components can be used in a product, they must go through the following processes:

System design
Determining the necessary structure and interconnections of the SoC-600 components that form the CoreSight debug and trace subsystem.

Configuration
Defining the memory map of the system and the functional configuration of the SoC-600 components.

Integration
Connecting the SoC-600 components together, and to the SoC memory system and peripherals.

Verification
Verifying that the CoreSight debug and trace subsystem has been correctly integrated to the processor or processors in your SoC.

Implementation
Using 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 SoC-600 components.

1.6 css600 component list

CoreSight SoC-600 components are provided as RTL blocks, the name of each one prefixed with `css600_`.

The following table lists the components and their versions.

Table 1-1 css600 component list

Name	Description	Version	Revision	IP-XACT Version
css600_ahbap	AHB Access Port	r0p0	0	r0p0_0
css600_apb3toapb4adapter	APB3 to APB4 adapter	r0p0	-	r0p0_0
css600_apb4toapb3adapter	APB4 to APB3 adapter	r0p0	-	r0p0_0
css600_apbap	APB Access Port	r0p0	0	r0p0_0
css600_apbasyncbridge	APB Asynchronous Bridge	r0p0	-	r0p0_0
css600_apbic	APB Interconnect	r0p0	-	r0p0_0
css600_apbpaddrdbg31adapter	APB PADDRDBG[31] Adapter	r1p0	-	r1p0_0
css600_apbrom	APB ROM Table	r0p0	0	r0p0_0
css600_apbrom_gpr	APB ROM GPR	r0p0	0	r0p0_0
css600_apbsyncbridge	APB Synchronous Bridge	r0p0	-	r0p0_0
css600_apv1adapter	Access Port v1 Adapter	r0p0	0	r0p0_0
css600_atbasyncbridge	ATB Asynchronous Bridge	r0p0	-	r0p0_0
css600_atbbuffer	ATB Trace Buffer	r0p0	-	r0p0_0
css600_atbdownsizer	ATB Downsizer	r0p0	-	r0p0_0
css600_atbfunnel	ATB Trace Funnel	r0p0	0	r0p0_0
css600_atbreplicator	ATB Trace Replicator	r0p0	0	r0p0_0
css600_atbsyncbridge	ATB Synchronous Bridge	r0p0	-	r0p0_0
css600_atbupsizer	ATB Trace Upsizer	r0p0	-	r0p0_0
css600_authasyncbridge	Authentication Asynchronous Bridge	r0p0	-	r0p0_0
css600_authreplicator	Authentication Replicator	r0p0	-	r0p0_0
css600_authsyncbridge	Authentication Synchronous Bridge	r0p0	-	r0p0_0
css600_axiap	AXI Access Port	r0p0	0	r0p0_0
css600_channelpulseasyncbridge	Channel Pulse Asynchronous Bridge	r0p0	-	r0p0_0
css600_channelpulsesyncbridge	Channel Pulse Synchronous Bridge	r0p0	-	r0p0_0
css600_channelpulsetochanneladapter	Channel Pulse to Channel Adapter	r0p0	-	r0p0_0
css600_channeltochannelpulseadapter	Channel to Channel Pulse Adapter	r0p0	-	r0p0_0
css600_cti	Cross Trigger Interface	r0p0	0	r0p0_0
css600_ctitostmadapter	CTI to STM Adapter	r0p0	-	r0p0_0
css600_ctm	Cross Trigger Matrix	r0p0	-	r0p0_0
css600_dp	Debug Port	r0p1	-	r0p1_0

Table 1-1 css600 component list (continued)

Name	Description	Version	Revision	IP-XACT Version
css600_dpabortasyncbridge	DP Abort Asynchronous Bridge	r0p0	-	r0p0_0
css600_dpabortreplicator	DP Abort Replicator	r0p0	-	r0p0_0
css600_dpabortsyncbridge	DP Abort Synchronous Bridge	r0p0	-	r0p0_0
css600_eventlevelasyncbridge	Event Level Asynchronous Bridge	r0p0	-	r0p0_0
css600_eventlevelsyncbridge	Event Level Synchronous Bridge	r0p0	-	r0p0_0
css600_eventpulseasyncbridge	Event Pulse Asynchronous Bridge	r0p0	-	r0p0_0
css600_eventpulsesyncbridge	Event Pulse Synchronous Bridge	r0p0	-	r0p0_0
css600_eventpulsetoeventadapter	Event Pulse to Event Adapter	r0p0	-	r0p0_0
css600_eventtoeventpulseadapter	Event to Event Pulse Adapter	r0p0	-	r0p0_0
css600_jtagap	JTAG Access Port	r0p0	0	r0p0_0
css600_jtagtoswjadapter	JTAG to SWJ Adapter	r0p0	-	r0p0_0
css600_ntsasyncbridge	Narrow Timestamp Asynchronous Bridge	r0p0	-	r0p0_0
css600_ntsdecoder	Narrow Timestamp Decoder	r0p0	-	r0p0_0
css600_ntsencoder	Narrow Timestamp Encoder	r0p0	-	r0p0_0
css600_ntsreplicator	Narrow Timestamp Replicator	r0p0	-	r0p0_0
css600_ntssyncbridge	Narrow Timestamp Synchronous Bridge	r0p0	-	r0p0_0
css600_swjic	SWJ Interconnect	r0p0	-	r0p0_0
css600_swjtojtagadapter	SWJ to JTAG Adapter	r0p0	-	r0p0_0
css600_tmc	Trace Memory Controller	r0p1	1	r0p1_0
css600_tsgen	Timestamp Generator	r0p0	0	r0p0_0
css600_tsintp	Timestamp Interpolator	r0p0	-	r0p0_0
css600_tsreplicator	Timestamp Replicator	r0p0	-	r0p0_0
css600_cortexm4integrationcs	Cortex-M4 PIL	r0p0	1	r0p0_0

Note

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

1.7 Product revisions

This section describes the differences in functionality between product revisions of the CoreSight SoC-600.

r0p0

First release of CoreSight SoC-600.

r1p0

Second release of CoreSight SoC-600.
Added new components:

- *Narrow Timestamp* (NTS) components. Timestamp interpolator.
- *Processor Integration Layer* (PIL) components.

Chapter 2
DAP components functional description

This chapter describes the functionality of the SoC-600.

It contains the following sections:

2.1 Debug Port

The `css600_dp` module implements the JTAG and Serial Wire Debug Port protocols, either of which 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:

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

2.2 Error response handing

CoreSight SoC-600 *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 **dbgen**|**niden** is LOW.
- A Secure memory access when **spiden**|**spniden** is LOW.

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

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

Abort 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.

Invalid transaction This error only applies to the AXI-AP when a memory access is attempted with an invalid combination of CSW.Cache and CSW.Domain fields.

Internal register access errors are not logged in the TRR but are always passed on the APB slave interface. An internal register access generates an error only when a register write is attempted after an abort while the CSW.TrInProg bit is set.

The register field 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:

- If CSW.ERRNPASS is programmed as 0, the memory access errors are passed back on the APB slave interface.
- If CSW.ERRNPASS is programmed as 1, the 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 two exceptions to this rule, and in both cases the error is always passed on the APB slave interface. Regardless of the status of the CSW.ERRNPASS bit:

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

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, new memory accesses are allowed only if TRR.ERR=0, that is no previous errors are logged. If CSW.ERRSTOP and TRR.ERR are both set, 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 APB read data for all transactions that generate an error is UNKNOWN.

The following table shows 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, 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.

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 this scenario happens, 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 Authentication interface signals change while a memory transfer is in progress, the MEM AP still completes the ongoing transfer normally. The new Authentication 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.

2.3 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. Access Ports and related infrastructure can be cascaded in a CoreSight system to any depth. This process allows any memory system to contain a window into another memory system with a maximum memory footprint of 8KB in the source memory system.

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:

- Single clock domain.
- An APB4 slave interface.
- An APB4 master interface.
- An authentication interface.
- An Q-Channel LPI for high-level clock management.
- Auto-incrementing TAR.
- Stalling accesses.
- Data access size:
  - 32 bits only.
- Endianness:
  - Little endian only.
- Error response.
- CoreSight Component base pointer register.

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.

2.4 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. Access Ports and related infrastructure can be cascaded in a CoreSight system to any depth. This process allows any memory system to contain a window into another memory system with a maximum memory footprint of 8KB in the source memory system.

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:

- Single clock domain.
- An APB4 slave interface.
- An AHB5 master interface.
- An authentication interface.
- Support for AHB5 TrustZone® signaling.
- A Q-Channel LPI for high-level clock management.
- Auto-incrementing TAR.
- Stalling accesses.
- Data access size:
  - 8 bits, 16 bits, or 32 bits.
- Endianness:
  - Little endian only.
- Error response.
- CoreSight Component base pointer register.

The AHB-AP does not support:

- BURST or SEQ transactions.
- Exclusive accesses.
- Unaligned transfers.

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.

2.5 AXI Access Port

The `css600_axiap` implements the MEM-AP architecture to directly connect to an AXI memory system. You can connect it to other memory systems using a suitable bridging component.

This section contains the following subsections:

2.5.1 AXI-AP features

AXI-AP implements the following list of features.

- Single clock domain.
- AXI4 interface support.
- Auto-incrementing TAR.
- Stalling accesses.
- Access size.
  - 8, 16, 32, or 64 bits.
- Endianness:
  - Little endian only.
- Error response.
- ROM table pointer register.
- Large physical address extension support:
  - 32 or 64 bits.
- AXI transfers.
  - Write and read transfers.
  - Burst size of 1 only.
  - No out-of-order transactions.
  - No multiple outstanding accesses.
  - Only aligned transfers are supported.
- ACE-Lite.
  - Limited set of commands to support coherency in the system.
  - All transactions to Non-shareable memory regions.
  - Limited subset of transactions to shareable memory regions.
  - For reads only. Supports the ReadOnce transaction type.
  - For writes only. Supports the WriteUnique transaction type.
  - Barrier transactions.

2.5.2 DAP transfer abort

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

2.5.3 Additional AXI error responses

The AXI-AP produces error responses for AXI-initiated and AP-initiated transfers.

AXI initiated error responses

An error response that is received on the AXI master interface propagates onto the Debug APB bus as the transfer is completed.

For 64-bit data transfer, a sequence of two reads or writes must be generated on the Debug APB bus for a single 64-bit access on the AXI interface. For reads, the first read request on the Debug APB bus sends a read request on the AXI interface. For writes, a write access is sent on the AXI interface only after two write requests are received on the Debug APB bus.

Therefore, an error response that is received for a read request is for the first read request on the Debug APB bus while an error response that is received for a write request is for the second write request on the Debug APB bus.

AP-initiated error response

AXI-AP reads after a 64-bit AXI read sequence is broken

Read requests from the Debug APB bus must access both BDx registers, or a consecutive pair of DAR registers forming an aligned 64-bit address. Read requests must access the lower-numbered register first. For a DRW register access, two write requests are required to get the entire 64-bit word from the AXI interface.

All other accesses, such as a read followed by a write access to the same or different registers, return an error response to the DP.

AXI-AP writes after a 64-bit write sequence is broken

Write requests from the Debug APB interface must access both BDx registers of the pair, or consecutive DAR registers forming an aligned 64-bit address, and must access the lower-numbered register first. For a DRW register access, two write requests are required to build a 64-bit packet as write data on the AXI interface.

All other accesses, such as a write followed by another read-write access to different registers, return an error response.

For example, after accessing the DRW register, the next access on the Debug APB bus must be a write to the DRW register. Any other access returns an error response.

Similarly, after accessing BD0, the next access must be a write to BD1. Any other access returns an error response.

Aborted AXI barrier transaction

It is possible to abort a barrier transaction that has not yet completed. When the abort request is generated, the Debug APB transaction is completed in the next cycle. However the CSW.TrInPrg bit remains set to indicate that the AXI interface is busy waiting to complete the transaction. While the AXI interface is busy, a read-write request to DRW, DAR, or BD*x* registers that results in a transaction on the AXI interface, causes the AXI-AP to return an error response to the DP.

2.5.4 AXI transfers

Features that are supported by the AMBA4 AXI-compliant Master Port.

- Bursts of single transfer.
- Master processes one transaction at a time in the order they are issued.
- No out-of-order transactions.
- No issuing of multiple outstanding addresses.

Burst length

The AXI-AP supports a burst length of one transfer only. **ARLEN[3:0]** and **AWLEN[3:0]** are always `0b0000`.

Burst size

Supported burst sizes are:

- 8-bit.
- 16-bit.
- 32-bit.
- 64-bit.

Burst type

ARBURST and **AWBURST** signals are always `0b01`.

Because only bursts of one transfer are supported, burst type has no meaning in this context.

Atomic accesses

AXI-AP supports normal accesses only.

ARLOCK and **AWLOCK** signals are always `0b00`.

Unaligned accesses

Unaligned accesses are not supported. Depending on the size of the transfers, addresses must be aligned. For example, for 16-bit transfers, addresses must be half-word aligned, for 32-bit word transfers, addresses must be word-aligned, and for 64-bit double-word transfers, addresses must be double-word aligned.

- For 16-bit half word transfers:
  - Base address `0x01` is aligned and **AxADDR[7:0]** = `0x00`.
  - Base address `0x02` is retained and **AxADDR[7:0]** = `0x02`.
- For 32-bit word transfers:
  - Base address `0x01` to `0x03` is aligned and **AxADDR[7:0]** = `0x00`.
  - Base address `0x04` is retained and **AxADDR[7:0]** = `0x04`.
- For 64-bit word transfers:
  - Base address `0x04` is aligned and **AxADDR[7:0]** = `0x00`.
  - Base address `0x08` is retained and **AxADDR[7:0]** = `0x08`.

2.5.5 Valid combinations of AxCACHE and AxDOMAIN

Valid combinations of **AxCACHE** and **AxDOMAIN**.

Table 2-1 Valid combinations of AxCACHE and AxDOMAIN values

AxCACHE[3:0]	Access type	AxDOMAIN	Domain type	Valid
`0b0000`	Device	`0b00`	Non-shareable	No
`0b0001`		`0b01`	Inner-shareable	No
		`0b10`	Outer-shareable	No
		`0b11`	System	Yes
`0b0010`	Non-Cacheable	`0b00`	Non-shareable	Enabled
`0b0011`		`0b01`	Inner-shareable	Enabled
		`0b10`	Outer-shareable	Enabled
		`0b11`	System	Yes
`0b010x`	-	-	-	No
`0b100x`	-	-	-	
`0b110x`	-	-	-	
`0b011x`	Write Through	`0b00`	Non-shareable	Yes
`0b101x`		`0b01`	Inner-shareable	Yes
`0b111x`	Write Back	`0b10`	Outer-shareable	Yes
		`0b11`	System	No

2.6 JTAG Access Port

The `css600_jtagap` provides JTAG access to on-chip components, operating as a JTAG master port to drive JTAG chains throughout a SoC.

The JTAG command protocol is byte-oriented, with a word wrapper on the read and write ports to yield acceptable performance from the 32-bit internal data bus in the DAP. Daisy chaining is avoided by using a port multiplexer. In this way, slower cores do not impede faster cores.

See the *ARM® Debug Interface Architecture Specification ADIv6.0* for more information.

2.7 Access Port v1 Adapter

Use the `css600_apv1adapter` to connect a legacy *Access Port* (AP) with a *DAP Internal* (DAPBus) slave interface into an CoreSight Architecture v3 system.

The `css600_apv1adapter`:

- Maps the legacy AP registers into the *ARM® CoreSight™ Architecture Specification v3.0* APB4 memory map.
- Provides ID registers that allow a debugger to identify the combination as a mapped legacy Access Port.
- Provides Integration registers that allow a debugger to check connectivity of the *DP Abort* signal.

2.8 DP Abort replicator

The `css600_dpabortreplicator` 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_dpabortreplicator`.

Use the `css600_dpabortreplicator` to connect a single DP Abort master interface to multiple DP Abort slave interfaces. You must connect the DP Abort output from the Debug Port to every Access Port that appears in the Debug Port memory space.

2.9 DP Abort asynchronous bridge

The `css600_dpabortasyncbridge` asynchronous bridge is a wrapper component that instantiates a pulse asynchronous bridge.

The bridge is used to transfer the **dp_abort** signal across a clock or power domain boundary. The **dp_abort** signal is a pulse event that is used to unlock a deadlocked transaction on a DP to AP interconnection.

2.10 DP Abort synchronous bridge

The `css600_dpabortsyncbridge` synchronous bridge is a wrapper component that instantiates a pulse synchronous bridge.

The bridge is used to transfer the **dp_abort** signal across a power domain boundary. The **dp_abort** signal is a pulse event that is used to unlock a deadlocked transaction on a DP to AP interconnection.

2.11 JTAG to SWJ adapter

The `css600_jtagtoswjadapter` 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_jtagtoswjadapter`.

Use the `css600_jtagtoswjadapter` to connect a JTAG master interface to a *Serial Wire/JTAG* (SWJ) slave interface. This might be necessary when connecting a Debug Port to the `css600_jtagap`.

2.12 SWJ to JTAG adapter

The `css600_swjtojtagadapter` is provided to support stitching in an IP-XACT tooling product.

Use the `css600_swjtojtagadapter` to connect a *Serial Wire/JTAG* (SWJ) master interface to a JTAG slave interface.

Chapter 3
APB infrastructure components functional description

This chapter describes the functionality of the APB infrastructure components.

It contains the following sections:

3.1 APB interconnect

The `css600_apbic` is used to provide connections between APB4 masters and APB4 slaves anywhere in a CoreSight system. APB4 masters might be debug ports, APB Access Ports, or other APB masters from a compute subsystem. It is a two-part meta-component that supports up to 4 slave interfaces and up to 64 master interfaces. The interconnect has the following features:

- Single clock domain.
- Decoder component configurable for up to 4 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.

This section contains the following subsections:

- *3.1.1 Arbitration* on page 3-38.
- *3.1.2 Error response* on page 3-38.

3.1.1 Arbitration

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

The algorithm that is used is when a slave interface raises a request, the highest priority is given 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.

3.1.2 Error response

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

An error response is returned under any of the following conditions:

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

3.2 APB ROM table

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

The `css600_apbrom_gpr` is a *ROM Table* that includes the *Granular Power Requestor* (GPR) function.

Use the `css600_apbrom` or `css600_apbrom_gpr` to:

- Identify part of your system or subsystem.
- Indicate the locations of other CoreSight components in the same address space to an *External Debugger*.
- Request power or reset to be supplied to components in the debug subsystem or the wider system (`css600_apbrom_gpr` only).

The `css600_apbrom` and `css600_apbrom_gpr` support up to 511 32-bit component entries, which are set by configuration parameters. The `css600_apbrom` and `css600_apbrom_gpr` support dynamic control of the *ROM Table* IDs and, optionally the presence of each entry, using configuration input signals. These features make the *ROM Table* suitable for use in configurable and hardened subsystems.

The `css600_apbrom_gpr` version adds the capability to request power or reset to individual components or parts of a system through a power or reset controller that is implemented outside the CoreSight subsystem. Power request interface numbers are normally aligned to power domain IDs configured into the *ROM Table*.

The GPR provides the following extra features:

- Authentication interface to control access to power and reset control features.
- Configurable number, up to 32, of debug power request interfaces, comprising a **cdbgpwrupreq** and **cdbgpwrupack** pair of signals.
- Configurable number, up to 32, of system power request interfaces, comprising a **csyspwrupreq** and **csyspwrupack** pair of signals.
- A debug reset request interface, comprising a **cdbgrstreq** and **cdbgrstack** pair of signals.
- A system reset request interface, comprising a **csysrstreq** and **csysrstack** pair of signals.

3.3 APB asynchronous bridge

The `css600_apbasyncbridge` asynchronous bridge 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.

3.4 APB synchronous bridge

The `css600_apbsyncbridge` synchronous bridge is used where an AMBA APB4 bus is required to cross a power domain boundary but not a clock domain boundary.

The APB asynchronous bridge provides the following features:

- Two synchronous clock domains with any frequency difference. The clocks must be skew balanced and from a common source, so that they are high-level-gated by a common control point.
- Two independent power domains, either of which can be switched relative to the other.
- Two Q-Channel LPIs for clock and one for power switching management.
- Two-part meta-component with separate slave and master side components.
- Configurable APB address width.

3.5 APB PADDRDBG31 Adapter

The `css600_apbpaddrdbg31adapter` module, enables you to integrate a CoreSight Architecture v2.0 component or subsystem into a CoreSight debug and trace subsystem (CSSYS).

Use the `css600_apbpaddrdbg31adapter` to integrate legacy components that have a dedicated **paddrdbg31** signal into the memory map of the ARM CoreSight SoC-600 CSSYS. The `css600_apbpaddrdbg31adapter`:

- Replaces the 2GB split at `0x80000000` with a user-defined split.
- Maps the two views of the component to consecutive regions of the memory map.
- Maps the external debugger view to the lower region.
- Maps the self-hosted view to the upper region.

3.6 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.

3.7 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.

Chapter 4
ATB infrastructure components functional description

This chapter describes the functionality of the ATB infrastructure components.

It contains the following sections:

4.1 ATB upsizer

The `css600_atbupsizer` module enables you to increase the data width of an AMBA Trace Bus.

Use the `css600_atbupsizer` when you connect an AMBA Trace Bus master interface to a wider AMBA Trace Bus slave interface.

4.2 ATB downsizer

The `css600_atbdownsizer` module enables you to reduce the data width of an AMBA Trace Bus.

Use the `css600_atbdownsizer` when you must connect an AMBA Trace Bus master interface to a narrower AMBA Trace Bus slave interface.

4.3 ATB funnel

The `css600_atbfunnel` is used when more than one trace source must be merged into a single trace stream.

The funnel is configurable for the number of slave interfaces, from 2-8, and comes in programmable or non-programmable configurations. The register map of the programmable version is described in the programmers model section.

The programmable configuration allows the following features:

- Independent enable control for each slave port.
- Independent priority setting for each slave port, so that higher priority ports are serviced ahead of lower priority ports.
- Programmable hold time to reduce input switching that is based on trace ID value.
- Registers to allow integration testing of the trace network.

4.4 ATB replicator

The `css600_atbreplicator` splits a single trace stream into two trace streams for systems that have more than one trace sink component.

An optional programmable configuration is available that provides the following features:

- Filtering of trace IDs to allow some IDs to go to master port 0 and some to master port 1.
- Registers to allow integration testing of the trace network.

4.5 ATB trace buffer

The `css600_atbbuffer` is used in situations where some local smoothing of trace bandwidth is required in a trace network.

The ATB trace buffer has the following features:

- Configurable trace data width up to 128 bits.
- Configurable buffer depth up to 256 entries.
- Configurable threshold for buffer fill level before starting to empty.

4.6 ATB asynchronous bridge

The `css600_atbasyncbridge` asynchronous bridge is used to transport the AMBA trace bus across a clock or power domain boundary.

The ATB 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.
- Two-part meta-component with separate slave and master side components.
- Configurable ATB data width.
- Configurable for 2- or 3-stage synchronizers.
- Automatically manages upstream flush of trace data before power down.

4.7 ATB synchronous bridge

The `css600_atbsyncbridge` synchronous bridge is used to transport the AMBA trace bus across a power domain boundary.

The ATB synchronous bridge provides the following features:

- Two synchronous clock domains with any frequency difference. The clocks must be skew balanced and from a common source so that they are high-level-gated by a common control point.
- Two independent power domains, either of which can be switched relative to the other.
- Two Q-Channel LPIs for clock and power switching management.
- Two-part meta-component with separate slave and master side components.
- Configurable ATB data width.
- Configurable for 2- or 3-stage synchronizers.
- Automatically manages upstream flush of trace data before power down.

4.8 Trace Memory Controller

The `css600_tmc` *Trace Memory Controller* (TMC) is used for capturing trace data into local memory, or the system memory. The trace can be read by an off-chip external debugger, or by on-chip self-hosted debug software.

The TMC can be configured into four different configurations. They are:

Embedded Trace Buffer (ETB) — Enables trace to be stored in a dedicated SRAM that is used as a circular buffer.

Embedded Trace FIFO (ETF) — Enables trace to be stored in a dedicated SRAM, used either as a circular buffer or as a FIFO. The functionality of the ETF configuration is a superset of the functionality of the ETB configuration. In a CoreSight system, the ETF can be inserted anywhere on the trace bus and used as a FIFO to smooth out a bursty trace.

Embedded Trace Router (ETR) — Enables trace to be routed over an AXI interface to the system memory or to any other AXI slave. The ETR configuration can be used for sending the trace off-chip through a streaming device such as a High *Speed Serial Trace Port* (HSSTP) link layer or through a functional I/O.

Embedded Trace Streamer (ETS) — Enables trace to be routed over an AXI4-Stream interface to a streaming device such as an HSSTP link layer, either directly or through an AXI4 Stream interconnect. ETS retains a subset of the ETR feature set, and removes those features not required for the streaming application, and provides a lower gate count than ETR uses for streaming.

The TMC can be programmed to capture trace in four different modes:

Circular Buffer mode — This mode is available in all four TMC configurations. TMC captures trace using its storage as a circular buffer, overwriting old trace when the buffer is full. No trace is output until capture is complete. In this mode, trace capture can automatically stop after receiving a trigger signal.

Hardware Read FIFO mode — This mode is available only in the ETF configuration. TMC uses its storage as a FIFO, acting as a link between a trace source and a trace sink. No trace is lost or overwritten, and backpressure is applied through the *AMBA Trace Bus* (ATB) to the trace source when the FIFO becomes full. Most trace sources eventually overflow when subject to backpressure for a long time, but it is always the trace sources that lose trace, not the ETF. This mode enables large bursts of trace to be smoothed, reducing the need to exert backpressure, so that:

- Trace can be output over a trace port using fewer pins than would be required without the ETF, by smoothing peaks in trace bandwidth over long periods.
- A subsequent ETR receiving trace through ETF can cope with large delays that are introduced by higher priority masters on the AXI interconnect without losing the trace.

Software Read FIFO mode 1 — Software Read FIFO mode 1: This mode is available in ETB, ETF, and ETR configurations. In this mode, the component functions as a FIFO where data is read out by software over Debug APB interface. This mode provides a low-speed communication channel for trace data, reusing the existing programming interface.

Software Read FIFO mode 2 — This mode is available only in ETR configuration. In this mode, the trace memory accessible through the AXI interface is used as a FIFO. The key difference from Software Read FIFO mode 1 is that the data is read out by debugger directly from the memory, bypassing the ETR datapath. The memory read pointer is still managed by the ETR so the debugger must inform the ETR of the amount of trace extracted. The trace that is extracted from the memory can be output over a streaming interface, such as USB, which provides much higher bandwidth than the debug APB interface.

This section contains the following subsections:

- *4.8.1 TMC register access dependencies* on page 4-54.

4.8.1 TMC register access dependencies

Not all TMC registers can be read and written under the same conditions.

Writes to TMC registers

You can only write to TMC registers under specific conditions.

The following table shows the conditions necessary to write to each TMC register. Writing to the TMC under conditions other than those listed results in UNPREDICTABLE behavior. An x indicates that any value is permitted.

Write accesses to TMC registers

Register	Name	Offset	CTL.TraceCaptEn	STS.TMCReady	MODE	ITCTRL.IME
RSZ (ETR)	RAM Size register	0x004	0	1	x	0
STS.MemErr (ETR)	Status Register	0x00C	x	1	x	0
STS.Full (ETR, ETS)	Status Register	0x00C	0	1	x	0
RRD	RAM Read Data Register	0x010	Read-only			
RRP	RAM Read Pointer Register	0x014	0	1	x	0
RWP	RAM Write Pointer Register	0x018	0	1	x	0
TRG	Trigger Counter Register	0x01C	0	1	x	0
CTL	Control Register	0x020	x	x	x	0
RWD	RAM Write Data Register	0x024	0	1	x	0
MODE	Mode Register	0x028	0	1	x	0
LBUFLEVEL	Latched Buffer Fill Level	0x02C	Read-only			
CBUFLEVEL	Current Buffer Fill Level	0x030	Read-only			
BUFWM	Buffer Level Water Mark	0x034	0	1	x	0
RRPHI	RAM Read Pointer High Register	0x038	0	1	x	0
RWPHI	RAM Write Pointer High Register	0x03C	0	1	x	0
AXICTL	AXI Control Register	0x110	0	1	x	0
DBALO	Data Buffer Address Low Register	0x118	0	1	x	0
DBAHI	Data Buffer Address High Register	0x11C	0	1	x	0

(continued)

Write accesses to TMC registers						
Register	**Name**	**Offset**	**CTL.TraceCaptEn**	**STS.TMCReady**	**MODE**	**ITCTRL.IME**
RURP	RAM Update Read Pointer Register	0x120	1	x	SWF2	0
FFSR	Formatter and Flush Status Register	0x300	Read-only			
FFCR.DrainBuffer (ETF)	Formatter and Flush Control Register	0x304	1	1	CB	0
FFCR.StopOnTrigEvt	Formatter and Flush Control Register	0x304	1	x	CB	0
			0	x	x	0
FFCR.StopOnFl	Formatter and Flush Control Register	0x304	x	x	x	0
FFCR.TrigOnFl	Formatter and Flush Control Register	0x304	x	x	x	0
FFCR.TrigOnTrigEvt	Formatter and Flush Control Register	0x304	1	x	CB	0
			0	x	x	0
FFCR.TrigOnTrigIn	Formatter and Flush Control Register	0x304	x	x	x	0
FFCR.FlushMan	Formatter and Flush Control Register	0x304	x	x	x	0
FFCR.FOnTrigEvt	Formatter and Flush Control Register	0x304	1	x	CB	0
			0	x	x	0
FFCR.FOnFlIn	Formatter and Flush Control Register	0x304	x	x	x	0
FFCR.EnTI	Formatter and Flush Control Register	0x304	0	1	x	0
FFCR.EnFt	Formatter and Flush Control Register	0x304	0	1	x	0
PSCR	Periodic Synchronization Counter Register	0x308	0	1	x	0
ITATBMDATA0 (ETF)	Integration Test ATB Master Data 0 Register	0xED0	x	x	x	1
ITATBMCTR2 (ETF)	Integration Test ATB Master Interface Control 2 Register	0xED4	Read-only			
ITATBMCTR1 (ETF)	Integration Test ATB Master Interface Control 1 Register	0xED8	x	x	x	1
ITATBMCTR0 (ETF)	Integration Test ATB Master Interface Control 0 Register	0xEDC	x	x	x	1

(continued)

Write accesses to TMC registers						
Register	**Name**	**Offset**	**CTL.TraceCaptEn**	**STS.TMCReady**	**MODE**	**ITCTRL.IME**
ITEVTINTR	Integration Test Event & Interrupt Status Register	0xEE0	x	x	x	1
ITTRFLIN	Integration Test Trigger In and Flush In Register	0xEE8	Read-only			
ITATBDATA0	Integration Test ATB Data 0 Register	0xEEC	Read-only			
ITATBCTR2	Integration Test ATB Control 2 Register	0xEF0	x	x	x	1
ITATBCTR1	Integration Test ATB Control 1 Register	0xEF4	Read-only			
ITATBCTR0	Integration Test ATB Control 0 Register	0xEF8	Read-only			
ITCTRL	Integration Mode Control Register	0xF00	0	1	x	x
CLAIMSET	Claim Tag Set Register	0xFA0	x	x	x	x
CLAIMCLR	Claim Tag Clear Register	0xFA4	x	x	x	x
AUTHSTATUS	Authentication Status Register	0xFB8	Read-only			
DEVARCH	Device Architecture Register	0xFBC	Read-only			
DEVID	Device Configuration Register	0xFC8	Read-only			
DEVTYPE	Device Type Identifier Register	0xFCC	Read-only			
PIDR4-7	Peripheral ID Registers 4-7	0xFD0-0xFDC	Read-only			
PIDR0-3	Peripheral ID Registers 0-3	0xFE0-0xFEC	Read-only			
CIDR0-3	Component ID Registers 0-3	0xFF0-0xFFC	Read-only			

Reads from TMC registers

You can only read from TMC registers under specific conditions.

The following table lists the conditions under which read accesses from TMC registers return valid values. Reads at other times return UNKNOWN values. An x indicates that any value is permitted.

Read accesses to TMC registers							
Register	**Name**	**Offset**	**CTL.TraceCaptEn**	**STS.TMCReady**	**MODE**	**ITCTRL**	**Remarks**
RSZ	RAM Size register	0x004	x	x	x	x	
STS.MemErr	Status Register	0x00C	x	x	x	0	
STS.Empty	Status Register	0x00C	1	x	x	0	
STS.FtEmpty	Status Register	0x00C	x	x	x	0	
STS.TMCReady	Status Register	0x00C	x	x	x	0	
STS.Triggered	Status Register	0x00C	1	x	CB	0	
			0	x	x	0	Value of this bit when trace capture stops is held.
STS.Full	Status Register	0x00C	x	x	x	0	Value of this bit when trace capture stops is held.
RRD	RAM Read Data Register	0x010	0	1	x	0	
			1	x	SWF1	0	If trace memory is empty, data returned is `0xFFFFFFFF`.
			1	1	CB	0	
RRP	RAM Read Pointer Register	0x014	1	x	SWF1, SWF2	0	
			1	1	CB	0	
			0	1	x	0	
RWP	RAM Write Pointer Register	0x018	1	x	SWF1, SWF2	0	
			1	1	CB	0	
			0	1	x	0	
TRG	Trigger Counter Register	0x01C	1	x	CB	0	The trigger counter is active only in Circular buffer mode.
			0	1	x	0	
CTL	Control Register	0x020	x	X	x	0	
RWD	RAM Write Data Register	0x024	Write-only				
MODE	Mode Register	0x028	1	x	x	0	

(continued)

Read accesses to TMC registers							
Register	**Name**	**Offset**	**CTL.TraceCaptEn**	**STS.TMCReady**	**MODE**	**ITCTRL**	**Remarks**
LBUFLEVEL	Latched Buffer Fill Level	0x02C	1	x	x	0	
			0	1	x	0	Value of this register when trace capture stops is held.
CBUFLEVEL	Current Buffer Fill Level	0x030	1	x	x	0	
			0	1	x	x	Value of this register when trace capture stops is held.
BUFWM	Buffer Level Water Mark	0x034	x	x	x	x	Programmed registers can be read at any time – the return value is the value that was programmed.
RRPHI	RAM Read Pointer High Register	0x038	1	x	SWF1, SWF2	0	
			1	1	CB	0	
			0	1	x	0	
RWPHI	RAM Write Pointer High Register	0x03C	1	0	SWF1, SWF2	0	
			1	1	CB	0	
			0	1	x	0	
AXICTL	AXI Control Register	0x110	x	x	x	x	
DBALO	Data Buffer Address Low Register	0x118	x	x	x	x	
DBAHI	Data Buffer Address High Register	0x11C	x	x	x	x	
RURP	RAM Update Read Pointer Register	0x120	Write-only				
FFSR	Formatter and Flush Status Register	0x300	x	x	x	0	

(continued)

Read accesses to TMC registers							
Register	**Name**	**Offset**	**CTL.TraceCaptEn**	**STS.TMCReady**	**MODE**	**ITCTRL**	**Remarks**
FFCR	Formatter and Flush Control Register	0x304	x	x	x	0	
PSCR	Periodic Synchronization Counter Register	0x308	x	x	x	0	
ITATBMDATA0	Integration Test ATB Master Data 0 Register	0xED0	Write-only				
ITATBMCTR2	Integration Test ATB Master Interface Control 2 Register	0xED4	x	x	x	1	
ITATBMCTR1	Integration Test ATB Master Interface Control 1 Register	0xED8	x	x	x	1	
ITATBMCTR0	Integration Test ATB Master Interface Control 0 Register	0xEDC	Write-only				
ITEVTINTR	Integration Test Event & Interrupt Status Register	0xEE0	Write-only				
ITTRFLIN	Integration Test Trigger In and Flush In Register	0xEE8	x	x	x	1	
ITATBDATA0	Integration Test ATB Data Register 0	0xEEC	x	x	x	1	
ITATBCTR2	Integration Test ATB Control 2 Register	0xEF0	Write-only				
ITATBCTR1	Integration Test ATB Control 1 Register	0xEF4	x	x	x	1	
ITATBCTR0	Integration Test ATB Control 0 Register	0xEF8	x	x	x	1	
ITCTRL	Integration Mode Control Register	0xF00	x	x	x	x	
CLAIMSET	Claim Tag Set Register	0xFA0	x	x	x	x	

(continued)

Read accesses to TMC registers							
Register	**Name**	**Offset**	**CTL.TraceCaptEn**	**STS.TMCReady**	**MODE**	**ITCTRL**	**Remarks**
CLAIMCLR	Claim Tag Clear Register	0xFA4	x	x	x	x	
AUTHSTATUS	Authentication Status Register	0xFB8	x	x	x	x	
DEVARCH	Device Architecture Register	0xFBC	x	x	x	x	
DEVID	Device Configuration Register	0xFC8	x	x	x	x	
DEVTYPE	Device Type Identifier Register	0xFCC	x	x	x	x	
PIDR4-7	Peripheral ID Registers 4-7	0xFD0-0xFDC	x	x	x	x	
PIDR0-3	Peripheral ID Registers 0-3	0xFE0-0xFEC	x	x	x	x	
CIDR0-3	Component ID Registers 0-3	0xFF0-0xFFC	x	x	x	x	

4.8.2 Clock and reset

The TMC has a single clock input **clk** and an active LOW reset input **reset_n**.

reset_n resets all interfaces and control registers except some of the memory mapped control registers. See the appropriate *Register summary* for your chosen TMC configuration for details of registers that are not initialized on reset and must be programmed before enabling TMC trace capture.

——— **Note** ———

When in ETR configuration, the TMC can be in a different power or reset domain from the AXI slave to which it is connected.

———————————

4.8.3 Interfaces

The TMC has the following interfaces:

- Debug APB interface.
- ATB slave interface.
- ATB master interface.
- Memory interface.
- AXI master interface.
- AXI stream master interface.
- Clock Q-Channel Low-Power Interface.
- Event interfaces.
- Buffer interrupt interface.
- Authentication interface.
- DFT interface.

The availability of each interface as a function of the configurations is shown in the following table:

Interfaces	ETB	ETF	ETR	ETS
Debug APB Interface	Present	Present	Present	Present
ATB Slave Interface	Present	Present	Present	Present
ATB Master Interface	Absent	Present	Absent	Absent
Memory Interface	Present	Present	Absent	Absent
AXI Master	Absent	Absent	Present	Absent
AXI Stream Master	Absent	Absent	Absent	Present
Clock Q-Channel	Present	Present	Present	Present
Event Interfaces	Present (5 event signals)	Present (5 event signals)	Present (5 event signals)	Present (5 event signals)
Buffer Interrupt	Absent	Absent	Present	Present
Authentication Interface	Absent	Absent	Present	Absent
DFT Interface	Present	Present	Present	Present

Debug APB interface

The Debug APB interface is used for programming the registers and to read the trace data from local SRAM or system AXI.

The Debug APB interface is compliant with the AMBA APB4 protocol.

ATB slave interface

The ATB slave interface is used to receive the trace data.

The interface can be connected to a replicator, a trace source, or any other component with a standard ATB master. The interface complies with the AMBA 4 ATB protocol specification.

ATB master interface

The ATB master interface is present only in ETF configuration and allows draining of trace data from local SRAM.

The interface can be connected to a replicator, trace sinks (ETB, or ETR), or any other component with a standard ATB slave. The interface complies with the AMBA 4 ATB protocol.

Memory interface

In ETB and ETF configurations, the memory interface supports access to on-chip SRAM to store and retrieve trace data.

AXI master interface

In ETR configuration, the AXI master interface replaces the memory interface that is used in ETB and ETF configurations.

The interface can be connected to an AXI interconnect for accessing system memory through a memory controller or can be connected to any other AXI slave in the system.

The AXI master interface supports up to 32 outstanding write transactions and zero outstanding reads. If an error response is returned at any time, the interface stops the operation until the debugger identifies and clears the error condition.

In ETR configuration, the memory size is programmable, rather than configurable. The width of the RSZ register determines the maximum size of the trace memory. The register is 31-bits wide, allowing a maximum value of `0x40000000`, representing 4GB. The trace memory can be located anywhere in the system address space with the start address aligned to a 4KB boundary. Some of the lower bits of AXI address buses **araddr_m** and **awaddr_m** are tied to 0 to ensure that all accesses are aligned to the AXI

data width. The number of bits to be tied to 0 is calculated as $\log_2$(`AXI_DATA_WIDTH`/8). For example, when the AXI data bus is 64-bits wide, the lower 3 bits of the address buses are tied to 0 to ensure that only 64-bit aligned accesses are used.

AXI stream master interface

In ETS configuration, the AXI stream master interface replaces the AXI interface that is used in ETR configuration.

The interface complies with the AMBA 4 AXI4-Stream protocol. It can be connected to any streaming device, such as an HSSTP link layer, either directly or through an AMBA AXI4-Stream interconnect for sending trace off-chip.

The ETS outputs only data bytes and no position or null bytes.

Clock Q-Channel Low-Power Interface

The TMC has a Q-Channel *Low-Power Interface* (LPI) for clock gating that is present in all four TMC configurations.

See the *AMBA® Low Power Interface Specification, ARM® Q-Channel and P-Channel Interfaces* for more information.

Event interfaces

The TMC has five event interfaces, comprising two slave and three master event signals, that can be connected to the *Cross Trigger Interface* (CTI).

Buffer interrupt interface

The TMC generates an interrupt that indicates that the trace memory holds an amount of trace data that is greater than, or equal to, the programmed watermark.

Authentication interface

The Authentication interface provides connections for the CoreSight Authentication Interface.

The Authentication interface is absent in the ETB, ETF, and ETS configurations, because the accesses are always non-invasive in those configurations and therefore the accesses do not have to be authenticated.

DFT interface

The DFT interface is a single active HIGH signal, **dftcgen**.

When asserted, **dftcgen** overrides the functional clock enable signal and forces the clock to remain ungated through the internal clock gate.

4.8.4 Operation

The state machine of the TMC.

Architectural state machine

The Trace Capture Enable bit, CTL.TraceCaptEn, and TMC Ready bit, STS.TMCReady, define the TMC states.

The operating states of Trace Memory Controller are:

DISABLED

DISABLED is the default state of TMC after reset and whenever CTL.TraceCaptEn is cleared. All programming must be performed in this state.

When the TMC is in DISABLED state, the contents of most registers, including the MODE and FFCR registers, have no effect. For backwards compatibility, the contents of the circular buffer can be read in this state. The debugger must manually manage the read pointer. The TMC enters RUNNING state from DISABLED state when the CTL.TraceCaptEn bit is set.

RUNNING

RUNNING is the functional state during which trace capture is performed.

The STOPPING state is entered from this state when a Stop event occurs.

STOPPING

In STOPPING state, the TMC begins to drain the trace data from its internal pipelines to the trace memory.

From the programmers model, the STOPPING state is indistinguishable from the RUNNING state. The STOPPED state is entered from STOPPING state when the following conditions are true:

- All trace has been output, including null padding, if necessary, to drain the last few bytes of trace, and the formatter and write buffer are empty.
- In Hardware Read FIFO mode, the FIFO and unformatter are also empty.
- In SWF1 and SWF2 modes, there must be space in the FIFO for data that is left in the pipeline to be written to the FIFO. To achieve this, it might be necessary to read extra data from the FIFO. If no space is available in the FIFO, then the STOPPED state is not reached.
- In ETS configuration, which is always in Circular Buffer mode, a STOPPING to STOPPED state transition depends on the rate at which data is accepted by the AXI4 Stream slave device.

STOPPED

In STOPPED state, no trace capture takes place, but data still in the trace memory can be read out.

When in STOPPED state:

- In Circular Buffer mode in ETB, ETF, and ETR configurations, except when streaming, the captured trace can be read out over debug APB.
- In Circular Buffer mode in ETF configuration, a drain can be initiated by setting the FFCR.DrainBuffer bit.
- In Software Read FIFO mode 1, the remaining contents of the FIFO can be read out over debug APB.
- In Software Read FIFO mode 2, a functional controller, such as USB, can directly read the remaining contents of the FIFO directly from the system memory.

The DISABLED state is entered from this state by clearing CTL.TraceCaptEn bit.

Note

The ETR configured for streaming and the ETS do not support reading of trace data over APB.

DRAINING

DRAINING state is only applicable in ETF configuration.

In DRAINING state, the contents of the buffer, that is captured in Circular Buffer mode, are drained over the ATB Master interface. The TMC returns to the STOPPED state when the buffer is empty.

DISABLING

DISABLING is an *emergency stop* state that can be entered at any time by clearing CTL.TraceCaptEn.

DISABLING state differs from the STOPPING state in the following ways:

- The TMC does not attempt to empty the contents of the FIFO in Hardware Read FIFO mode. Trace that is not yet output on the ATB Master interface is lost.
- The next transition is to the DISABLED state, not the STOPPED state. This transition means that:

  - In Circular Buffer mode, while the trace can still be read over the APB in ETB, ETF, and ETR non-streaming configurations, the ETF configuration drain operation is not possible.
  - In SWF1 and SWF2 modes, unretrieved trace is lost.
- Exit from DISABLING state is not dependent on reads performed from the RRD register, in SWF1 and SWF2 modes, or the ATB Master interface accepting writes in HWF mode.

  If the FIFO is full, then existing data is overwritten to enable the STOP sequence to complete. If a memory error occurs in DISABLING state, or if the STS.MemErr bit is already set and a hard stop occurs, more AXI writes are not performed. If this situation happens, the TMC discards the trace that is not output and directly moves from DISABLING to DISABLED state.

ARM recommends that the trace capture is stopped by programming an appropriate STOP event in the FFCR register. For example, trace capture can be stopped by setting the FFCR.StopOnFl bit, and then initiating a manual flush by setting the FFCR.FlushMan bit.

The *emergency stop* option is provided so that the TMC is programmer-compatible with the Classical ETB, where the only way to stop trace capture was by clearing CTL.TraceCaptEn bit. Use of the emergency stop is otherwise discouraged, especially in FIFO modes, where it can lead to loss of trace or even trace corruption.

In ETR and ETS configurations, transition from DISABLING to DISABLED state is delayed indefinitely if the AXI or AXI Stream interface is stalled indefinitely. In this case, you must clear the source of the stall, since it is not possible to abort a transfer when it has started.

Chapter 5
Timestamp components functional description

This chapter describes the functionality of the timestamp components.

It contains the following sections:

5.1 Timestamp generator

The `css600_tsgen` timestamp generator is used to generate a 64-bit rolling time for distribution to other CoreSight components that are used to align trace information.

5.2 Timestamp replicator

The `css600_tsreplicator` is an IP-XACT *Phantom Component* that is provided to support stitching in an ARM, or third-party, IP-XACT tooling product. There is no Verilog module for `css600_tsreplicator`.

Use the `css600_tsreplicator` to connect a single *Wide Timestamp* (WTS) master interface to multiple WTS slave interfaces. This is useful when you distribute WTS to multiple slaves in the same clock domain without the additional logic cost of a *Narrow Timestamp* (NTS) solution, and where the wire count of WTS is acceptable.

5.3 Timestamp interpolator

The timestamp interpolator increases the resolution of a timestamp.

The interpolator shifts the input timestamp left by 8 bits, and uses the extra low-order bits to provide a more accurate timestamp value. The greater accuracy is achieved by monitoring changes to the input timestamp value over time to predict how fast it counts.

This section contains the following subsections:

5.3.1 Clock and reset

The clock and reset signals of the timestamp interpolator are **clk** and **reset_n**.

clk
: Clock.

reset_n
: Active-LOW reset. **reset_n** is asynchronously asserted and must be synchronously deasserted.

5.3.2 Functional interface

The timestamp interpolator adjusts to changes in the rate of the incoming timestamp.

The interpolator ensures that the interpolated timestamp never counts backwards, and pauses incrementing the interpolated timestamp if it gets ahead of the input timestamp value.

5.3.3 Low-Power Interface

The timestamp interpolator has a Low-Power Interface to manage power reduction using high-level clock gating.

If the clock to the interpolator must be gated off, then the clock controller must use the *Low-Power Interface* (LPI). When the interpolator exits the low-power state, it automatically recalculates the interpolation ratio before advancing the interpolated timestamp.

5.3.4 Limitations

Use of the timestamp interpolator is subject to some limitations.

The limitations are:

- The timestamp interpolator must not be used in the timestamp network that is used to distribute processor time.
- There must be only one timestamp interpolator between the timestamp generator and a component that receives the timestamp.

5.4 Narrow timestamp asynchronous bridge

The `css600_ntsasyncbridge` narrow timestamp asynchronous bridge enables the transfer of timestamp information across asynchronous clock and power domains.

The narrow timestamp asynchronous bridge has the following key features:

- Supports asynchronous clock domain crossing.
- Narrow timestamp master and slave interfaces.
- Three LPIs for slave and master clock and power management.

The following figure shows the external connections on the narrow timestamp asynchronous bridge.

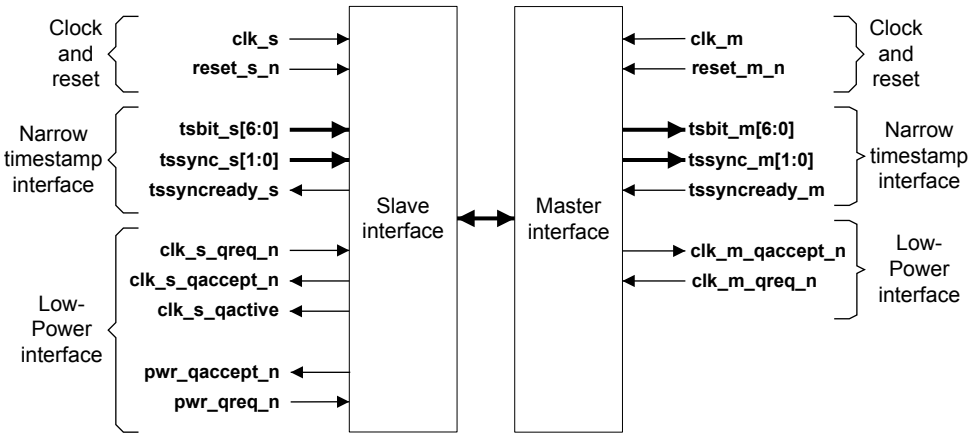

Figure 5-1 Narrow timestamp asynchronous bridge block diagram

This section contains the following subsections:

- *5.4.1 Operation* on page 5-69.
- *5.4.2 Low-power features* on page 5-69.
- *5.4.3 Timestamp protection from slow clock* on page 5-70.

5.4.1 Operation

The bridge implements an internal buffer to pass timestamp messages between the two clock domains, so that the timestamp resolution is maintained.

When the master interface of the bridge is running at a slower clock speed to the slave interface, the bridge discards some timestamp packets. The bridge ensures that the packets that remain convey the same time information, but incrementing in larger steps with lower resolution.

5.4.2 Low-power features

The narrow timestamp asynchronous bridge supports two power domains. Before a domain is powered down, the power controller must ensure that the bridge is in a safe state by using the Low-Power Interface.

Failure to ensure that the bridge is in a safe state might:

- Cause the rest of the timestamp interconnect to behave incorrectly.
- Corrupt timestamp values to be returned when the domain is powered up again.

When a low-power request is issued to the bridge by driving **pwr_qreq_n** LOW, the bridge:

- Drives **tssyncreadys** HIGH so that the clock can be stopped without affecting the rest of the system.
- Empties the internal buffer of timestamp messages. The clock of components that are connected to the master interface must still be running to enable these messages to be received.
- Brings the internal logic of the bridge to a safe state for one side to be powered down without the other.

When a powerup request is issued to the bridge by driving **pwr_qreq_n** HIGH, the bridge:

- Resynchronizes to the current timestamp value.
- Sends a resynchronization event through the narrow timestamp master interface so that downstream components synchronize to the correct timestamp value.

The bridge does not implement an automatic wake and therefore does not have a **qactive** signal on the power LPI. It always accepts requests to power down.

Each clock has an LPI for managing power reduction using high-level clock gating. If other components in the same part of the timestamp network are clock gated, the master side or the slave side can also be clock gated.

Slave side clock gating

Before gating the clock, the **clk_s_qreq_n** signal must be asserted LOW and the clock controller must wait for **clk_s_qaccept_n** to go LOW. While clock gated, the bridge ignores all timestamp packets arriving on the slave interface. If a request is made for a power state change using **pwr_qreq_n** (powering the master side up or down), then the slave clock is requested using the **clk_s_qactive** signal.

Master side clock gating

Before gating the clock, the **clk_m_qreq_n** signal must be asserted LOW and the clock controller must wait for **clk_m_qaccept_n** to go LOW. The clock can then be gated off indefinitely. While gated, the slave side buffer fills and it stalls the synchronization channel using **tssyncready_s**. When exiting the clock gated state, the bridge issues a resynchronization packet on the master interface.

5.4.3 Timestamp protection from slow clock

The bridge is designed to ensure that there is a limit to the deviation in output timestamps when compared to the input.

When the master interface of the bridge is running at a slower clock speed than the slave interface, some values of timestamp are discarded in the bridge. The master interface might always be running at a slower frequency than the slave interface. For example, if the timestamp generator is clocked at a higher frequency than the timestamp destination. Alternatively the master interface might normally be running at a higher frequency than the slave interface, but is occasionally stopped for power-saving purposes. In both of these scenarios, some values of the timestamp are discarded in the bridge.

The bridge ensures that the deviation is limited to the clock ratio rounded up to the next power of 2. For example, for a slave:master clock ratio of 10:1, where the slave interface is running ten times faster than the master interface, the output timestamps are no more than 16 timestamps lower than the input timestamps.

In situations where the master interface clock is very slow compared to the slave interface clock, the bridge has a mechanism to force an automatic resynchronization if the deviation gets beyond a predetermined limit. This limit is set in the `THRESHOLD` configuration option.

See the *ARM® CoreSight™ SoC-600 Configuration and Integration Manual* for a description of how to set the threshold.

5.5 Narrow timestamp synchronous bridge

The `css600_ntssyncbridge` narrow timestamp synchronous bridge enables the transfer of timestamp information across synchronous clock and power domains that have individual clock enables.

The narrow timestamp synchronous bridge has the following key features:

- Supports synchronous clock domain crossing:
  - SYNC 1:1.
  - SYNC 1:n.
  - SYNC n:1.
  - SYNC n:m.
- Narrow timestamp master and slave interfaces.
- Two Low-Power Interfaces for clock and power management.

The following figure shows the external connections on the narrow timestamp synchronous bridge.

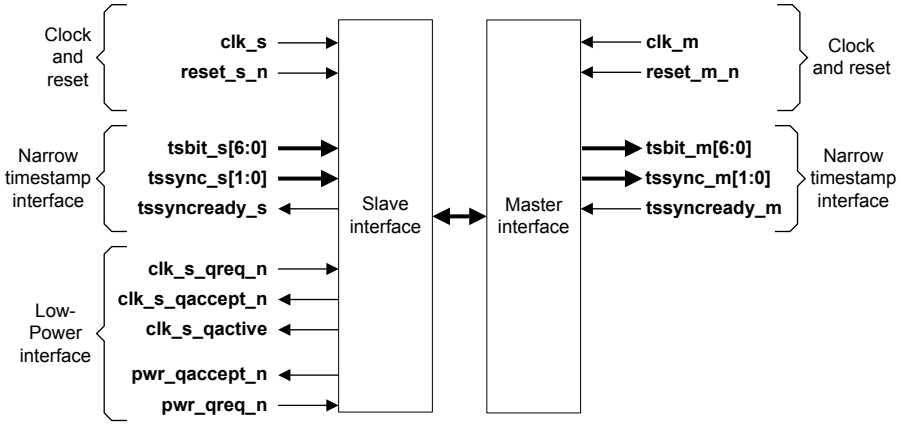

Figure 5-2 Narrow timestamp synchronous bridge block diagram

This section contains the following subsections:

- *5.5.1 Operation* on page 5-71.
- *5.5.2 Low-power features* on page 5-71.

5.5.1 Operation

The bridge implements a small internal buffer to pass timestamp messages between the two sides, so that the timestamp resolution is maintained.

When the master interface of the bridge is running at a slower clock speed to the slave interface, the bridge discards some timestamp packets. The bridge ensures that the packets that remain convey the same time information, but incrementing in larger steps with lower resolution.

5.5.2 Low-power features

The narrow timestamp asynchronous bridge supports a power domain boundary on the slave interface, outside the bridge.

Failure to ensure that the bridge is in a safe state might:

- Cause the rest of the timestamp interconnect to behave incorrectly.
- Corrupt timestamp values to be returned when the domain is powered up again.

When a low-power request is issued to the bridge by driving **pwr_qreq_n** LOW, the bridge:

- Drives **tssyncreadys** HIGH so that the clock can be stopped without affecting the rest of the system.
- Empties the internal buffer of timestamp messages. The clock of components that are connected to the master interface must still be running to enable these messages to be received.
- Brings the internal logic of the bridge to a safe state for one side to be powered down without the other.

When a powerup request is issued to the bridge by driving **pwr_qreq_n** HIGH, the bridge:

- Resynchronizes to the current timestamp value.
- Sends a resynchronization event through the narrow timestamp master interface so that downstream components synchronize to the correct timestamp value.

The bridge does not implement an automatic wake and therefore does not have a **qactive** signal on the power LPI. It always accepts requests to power down.

The clock has an LPI for managing power reduction using high-level clock gating. If other components in the same part of the timestamp network are clock gated, the bridge can also be independently clock gated.

The clocks have an LPI for managing power reduction using high-level clock gating. The master side and the slave side must be clocked from a single source so that when clock gated, both sides of the bridge are inactive.

Before gating the clock, the **clk_s_qreq_n** signal must be asserted LOW and the clock controller must wait for **clk_s_qaccept_n** to go LOW. While clock gated, the bridge ignores all timestamp packets arriving on the slave interface. If a request is made for a power state change using **pwr_qreq_n** to power the master side up or down, then the clock is requested using the **clk_s_qactive** signal.

5.6 Narrow timestamp decoder

The `css600_ntsdecoder` narrow timestamp decoder converts the narrow timestamp interface and synchronization data back to a 64-bit value, as required by CoreSight trace components.

The component decodes the narrow timestamp interface to a 64-bit wide timestamp signal. After reset, it outputs a value of zero until it has synchronized to the correct timestamp value.

The following figure shows the external connections on the timestamp decoder.

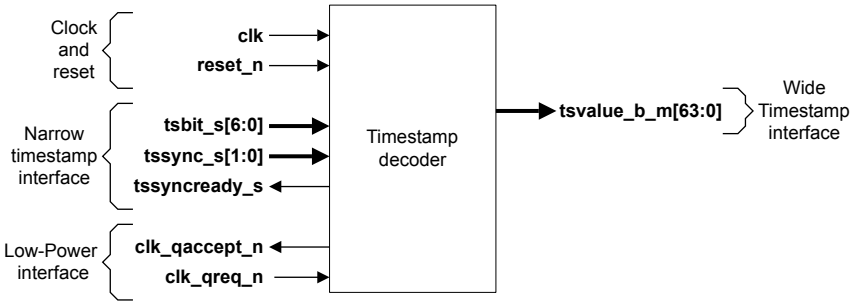

Figure 5-3 Timestamp decoder block diagram

5.7 Narrow timestamp encoder

The `css600_ntsencoder` narrow timestamp encoder converts the 64-bit timestamp value from the timestamp generator to a 7-bit encoded value, called a narrow timestamp.

The narrow timestamp encoder also encodes and sends the timestamp value over a 2-bit synchronization channel.

The following figure shows the external connections on the narrow timestamp encoder.

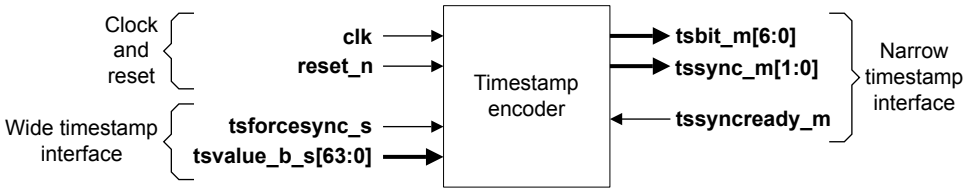

Figure 5-4 Narrow timestamp encoder block diagram

5.8 Narrow timestamp replicator

The `css600_ntsreplicator` narrow timestamp replicator distributes the encoded timestamp and synchronization data to multiple master interfaces.

The narrow timestamp replicator has the following key features:

- 1:n distribution of narrow timestamp bus.
- Configurable number of narrow timestamp master interfaces between 2 and 16.

The following figure shows the external connections on the narrow timestamp replicator.

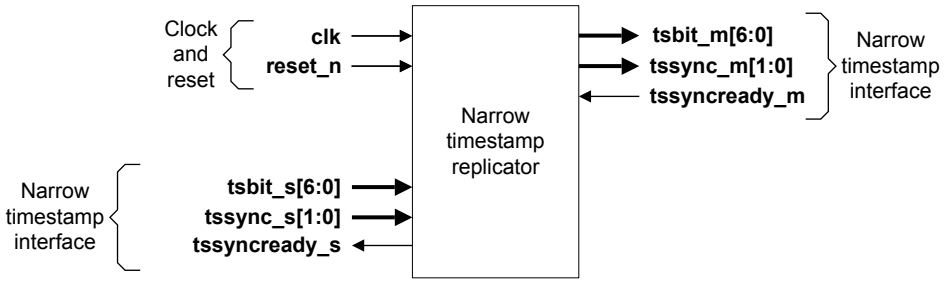

Figure 5-5 Narrow timestamp replicator block diagram

Chapter 6
Embedded Cross Trigger components functional description

The cross-triggering components enable CoreSight components to broadcast events between each other.

Events are distributed as follows:

- Each event type is connected to a trigger input on a *Cross Trigger Interface* (CTI).
- Each CTI can be programmed to connect each trigger input to each of four channels. If programmed to do so, when an input event occurs, it causes an event on the corresponding channel.

CTIs are connected to each other using one or more *Cross Trigger Matrices* (CTMs), through channel interfaces. When an event occurs on a channel, it is broadcast on that channel to all other CTIs in the system.

Each CTI can be programmed to connect each channel to each of several trigger outputs. If programmed to do so, when a channel event occurs, it causes an event on the trigger output.

Each CTI trigger output can be connected to a CoreSight component event input.

Cross triggering can take place between trigger inputs and outputs on a single CTI, or between multiple CTIs. CTIs can be programmed not to broadcast events for selected channels, so that certain events can only trigger output events on the same CTI. Only the CTIs are programmable.

It contains the following sections:

6.1 Event signaling protocol

The cross-triggering system does not attempt to interpret the events that are signaled through it.

Events are broadcast as a pulse. When an event passes across a clock domain boundary, using an asynchronous bridge, handshaking occurs to ensure that the event lasts for exactly one clock cycle in the destination clock domain.

Each channel is a shared broadcast medium, that can carry events from multiple sources, going to multiple domains. When a CTI sends events onto a channel it can coincide with other events on the same channel, so that the events become pulses of more than one clock cycle. This behavior is expected within the cross trigger system.

In usage models that count events that are passed through the cross-triggering system, events that occur close together might be merged into a single event with a single pulse when passed to another clock domain.

6.2 Cross Trigger Interface

The `css600_cti` Cross Trigger Interface connects one or more event sources and one or more event destinations to the cross trigger network.

The CTI has the following functional interfaces:

- Up to 32 trigger inputs, enabling events to be signaled to the CTI.
- Up to 32 trigger outputs, enabling the CTI to signal events to other components.
- A channel interface for connecting CTIs together using one or more CTMs.
- An APB interface for accessing the registers of the CTI.
- An Authentication interface for controlling access to certain debug events.
- Eight **asicctrl** signals that can be used to control external multiplexers.

The CTI includes configuration tie-off inputs that enable several trigger input and output types to be connected. For more information on configuring the trigger inputs and outputs, see the *ARM® CoreSight™ SoC-600 Configuration and Integration Manual*.

ARM recommends that the CTI that is connected to a processor is disabled before the processor clock is stopped. This operation minimizes the likelihood of unexpected events entering the cross-triggering system or affecting the processor when its clock is restarted. The CTICONTROL.CTIEN register bit can be used to globally disable the CTI without changing the event mapping programming.

6.2.1 asicctrl

The **asicctrl** output of the CTI can be used to control multiplexing on a CTI event input if necessary.

The exact configuration of any external multiplexing is user-defined. ARM does not define any relationship between values that are written to the control register and the actual configuration of the multiplexers.

Use the `EXT_MUX_NUM` parameter, to set a user-defined value that identifies the configuration of the multiplexers. See *External Multiplexer Control register, ASICCTRL* on page 9-674 for more information.

The following figure shows two of the **asicctrl** signals controlling multiplexers that are providing input to an **event_in** port in the CTI.

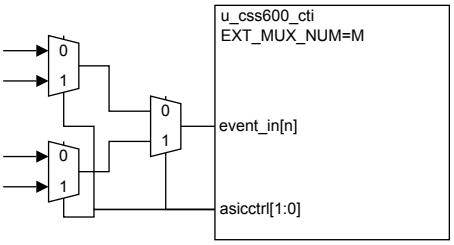

Figure 6-1 asicctrl signals controlling multiplexers

6.3 Cross Trigger Matrix

The `css600_ctm` Cross Trigger Matrix is used to connect CTI components together in a cross trigger system.

The component is configurable for up to 33 channel interfaces. If more than 33 CTIs must be connected together, then CTMs can be connected together without limitation.

6.4 Event Pulse to Event adapter

The `css600_eventpulsetoeventadapter` Event Pulse to Event adapter is a wrapper component that instantiates a slave half of a pulse async bridge with a configurable signal width.

The component provides the **req**/**ack** handshake that is required to interface a SoC-600 event to a legacy CTI, such as one in a CoreSight SoC-400 system.

6.5 Event to Event Pulse adapter

The `css600_eventtoeventpulseadapter` Event to Event Pulse adapter is a wrapper component that instantiates a master half of a pulse async bridge with a configurable signal width.

The component provides the **req**/**ack** handshake that is required to interface a legacy event source to a SoC-600 CTI.

6.6 Event Level asynchronous bridge

The `css600_eventlevelasyncbridge` Event Level asynchronous bridge is a wrapper component that instantiates a synchronizer.

The bridge is used to pass an event that operates as a level, rather than a pulse, across a clock domain boundary. The bridge can be used, for example, when using the software handshake configuration of a CTI event output, and the resulting event output must cross a clock or power domain boundary to reach its destination.

If more than one signal is to be transported across the same boundary, then the component can be configured for width.

6.7 Event Level synchronous bridge

The `css600_eventlevelsyncbridge` Event Level synchronous bridge is a wrapper component that instantiates a register slice as an aid to timing closure on events that must travel a long distance on chip.

If more than one signal is to be transported across the same boundary, then the component is configurable for width.

6.8 Event Pulse asynchronous bridge

The `css600_eventpulseasyncbridge` Event Pulse asynchronous bridge is used where an event signal, or a group of events, must cross a clock or power domain boundary.

The bridge has 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.
- Two-part meta-component with separate slave and master side components.
- Configurable up to 32-bits wide for transporting multiple events across the same boundary.

6.9 Event Pulse synchronous bridge

The `css600_eventpulsesyncbridge` Event Pulse synchronous bridge is used where an event signal, or a group of events, must cross a power domain boundary.

The bridge has the following features:

- Two synchronous clock domains with any frequency difference. The clocks must be skew balanced and from a common source so that they are high-level-gated by a common control point.
- Two independent power domains, either of which can be switched relative to the other.
- Two Q-Channel LPIs for clock and power switching management.
- Two-part meta-component with separate slave and master side components.
- Configurable up to 32-bits wide for transporting multiple events across the same boundary.

6.10 Channel Pulse to Channel adapter

The `css600_channelpulsetochanneladapter` Channel Pulse to Channel adapter is a wrapper component that instantiates a slave half of a pulse async bridge with a 4-bit signal width.

The component provides the **req**/**ack** handshake that is required to interface a SoC-600 CTI or CTM to a legacy CTI or CTM such as one in a CoreSight SoC-400 system.

6.11 Channel to Channel Pulse adapter

The `css600_channeltochannelpulseadapter` Channel to Channel Pulse adapter is a wrapper component that instantiates a master half of a pulse async bridge with a 4-bit signal width.

The component provides the **req**/**ack** handshake that is required to interface a SoC-600 CTI or CTM to a legacy CTI or CTM such as one in a CoreSight SoC-400 system.

6.12 Channel Pulse asynchronous bridge

The `css600_channelpulseasyncbridge` Channel Pulse asynchronous bridge is a wrapper component that instantiates a pulse async bridge with a 4-bit signal path.

The bridge is used to connect a CTI to a CTM, or two CTMs, where the signals must cross a clock or power domain boundary. The bridge has 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.
- Two-part meta-component with separate slave and master side components.

6.13 Channel Pulse synchronous bridge

The `css600_channelpulsesyncbridge` Channel Pulse synchronous bridge is a wrapper component that instantiates a pulse async bridge with a 4-bit signal path.

The bridge is used to connect a CTI to a CTM, or two CTMs, where the signals must cross a power domain boundary. The bridge has the following features:

- Two synchronous clock domains with any frequency difference. The clocks must be skew balanced, and from a common source, so that they are high-level-gated by a common control point.
- 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.
- Two-part meta-component with separate slave and master side components.

6.14 CTI to STM adapter

The `css600_ctitostmadapter` CTI to STM adapter is used to adapt a single event signal to two event inputs of a System Trace Macrocell.

Chapter 7
Authentication components functional description

This chapter describes the functionality of the authentication components.

It contains the following sections:

7.1 Authentication replicator

The `css600_authreplicator` is an IP-XACT *Phantom Component* that is provided to support stitching in an ARM, or third-party, IP-XACT tooling product. There is no Verilog module for `css600_authreplicator`.

Use the `css600_authreplicator` to connect a single Authentication master interface to multiple Authentication slave interfaces.

7.2 Authentication asynchronous bridge

The `css600_authasyncbridge` authentication asynchronous bridge is used where the Authentication interface must cross a clock or power domain boundary.

The bridge contains synchronizers to capture the signals in the receiving clock domain.

7.3 Authentication synchronous bridge

The `css600_authsyncbridge` authentication synchronous bridge is a register slice to aid timing closure for authentication signals that are crossing a large distance across a chip.

Chapter 8
Processor Integration Layer components

This chapter gives an overview of the Cortex-M4 *Processor Integration Layer* (PIL).

It contains the following sections:

8.1 Cortex-M4 PIL overview

A description of the Cortex-M4 *Processor Integration Layer* (PIL).

The Cortex-M4 PIL consists of the following:

- A processor that has an *Instrumentation Trace Macrocell* (ITM) and *Advanced High-performance Bus* (AHB)-*Access Port* (AP).
- An optional *Wakeup Interrupt Controller* (WIC).
- A ROM table that connects to the processor through a *Private Peripheral Bus* (PPB).
- An *Embedded Trace Macrocell* (ETM) trace unit that connects to the processor.
- A CTI for debug event communication.

The Cortex-M4 PIL supports the following external interfaces:

- AHB-Lite interfaces:
  - I-Code.
  - D-Code.
  - System.
- Two *Advanced Trace Bus* (ATB) interfaces that connect to the CoreSight subsystem.
- An *Advanced Peripheral Bus* (APB) interface for adding debug components to the PPB.
- An APB interface that connects to the debug port in the CoreSight subsystem.
- Processor-specific signals such as interrupt signals, system control signals, and status signals.

The following figure shows a block diagram of the Cortex-M4 PIL.

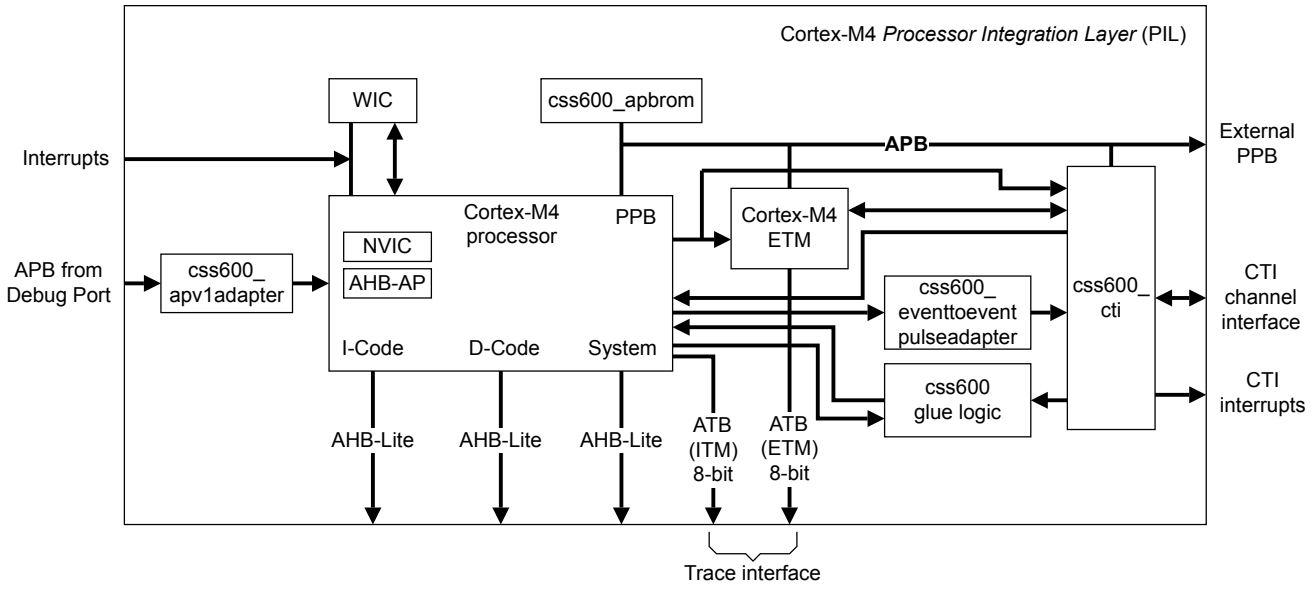

Figure 8-1 Cortex-M4 PIL block diagram

8.2 Cortex®-M4 PIL CoreSight component identification

CoreSight components have several IDs that identify the components.

The following table lists the CoreSight ID register reset values for the components present within the Cortex-M4 PIL. See the *ARM® CoreSight™ Architecture Specification v3.0* for information on the CoreSight ID scheme.

Table 8-1 Cortex-M4 PIL CoreSight ID register reset values

PID	CID	DevType	DevArch	Revision	Component
0x00000004000bb9e5	0xb105900d	0x00	0x47700a47	r0p0	css600_apv1adapter
0x00000004001bb4c6	0xb105900d	0x00	0x47700af7	r0p0	css600_cortexm4integrationcs ROM Table
0x00000004000bb000	0xb105e00d	0x00	0x00000000	r0p0	ARM v7M *System Control Space* (SCS)
0x00000004003bb002	0xb105e00d	0x00	0x00000000	r0p0	ARM v7M *Data Watchpoint and Trace* (DWT)
0x00000004002bb003	0xb105e00d	0x00	0x00000000	r0p0	ARM v7M *FlashPatch and Breakpoint* (FPB)
0x00000004003bb001	0xb105e00d	0x00	0x00000000	r0p0	ARM v7M *Instrumentation Trace Macrocell* (ITM)
0x00000004000bb925	0xb105900d	0x13	0x00000000	r0p0	Cortex-M4 ETM
0x00000004000bb9ed	0xb105900d	0x14	0x47701a14	r0p0	css600_cti

8.3 Cortex®-M4 PIL debug memory map

The debug components in the Cortex-M4 PIL share memory space with the processor system. Part of the system memory is allocated to the *Private Peripheral Bus* (PPB).

The following tables show the locations of the Cortex-M4 PIL CoreSight components.

Table 8-2 External PPB division

Address range	Components
0xE0041000-0xE0041FFF	ETM trace unit.
0xE0042000-0xE0042FFF	CTI.
0xE00FF000-0xE00FFFFF	ROM table.
0xE0040000-0xE0040FFF	External PPB expansion bus. In a standard single processor Cortex-M4 system, the Cortex-M4 TPIU uses this space.
0xE0043000-0xE00FEFFF	External PPB expansion bus.

Table 8-3 Internal PPB division

Address range	Section	Components
0xE0000000-0xE003FFFF	Internal PPB	These components are: • *Instrumentation Trace Macrocell* (ITM). • *Data Watchpoint and Trace* (DWT). • *Flash Patch and Breakpoint* (FPB). • *System Control Space* (SCS) including for example: — *Nested Vectored Interrupt Controller* (NVIC). — SysTick. — *Memory Protection Unit* (MPU).
0xE0040000-0xE00FFFFF	External PPB	These components are: • ROM table. • *Embedded Trace Macrocell* (ETM) trace unit. • *Cross Trigger Interface* (CTI).

Chapter 9
Programmers model

This chapter describes the programmers models for all CoreSight SoC-600 components that have programmable registers.

It contains the following sections:

9.1 css600 components programmers model

The following information applies to the css600 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 from the 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.

9.2 css600_dp introduction

This section describes the functions and 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*. Since the DP only supports 32-bit addressing, any read to BASEPTR1 always returns 0 and any writes to SELECT1 register are ignored.

This section contains the following subsections:

9.2.1 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 SELECTR.DPBANKSEL. The combinations of address offset and SELECT.DPBANKSEL value, and whether the register is accessible by the JTAG-AP, 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 9-1 css600_dp register summary

Offset	DPBANKSEL	Name	JTAG-DP	SW-DP	Reset	Description
`0x0000`	X	ABORT	No	Yes	`0x0c013477`	*AP Abort Register, ABORT* on page 9-103
`0x0000`	`0x0`	DPIDR	Yes	Yes	`0x0c013477`	*Debug Port Identification Register, DPIDR* on page 9-104
`0x0000`	`0x1`	DPIDR1	Yes	Yes	`0x0000008c`	*Debug Port Identification Register 1, DPIDR1* on page 9-105
`0x0000`	`0x2`	BASEPTR0	Yes	Yes	`0x00000032`	*Base Pointer Register 0, BASEPTR0* on page 9-106
`0x0000`	`0x3`	BASEPTR1	Yes	Yes	`0x00000000`	*Base Pointer Register 1, BASEPTR1* on page 9-107
`0x0004`	`0x0`	CTRLSTAT	Yes	Yes	`0x00000000`	*Control/Status Register, CTRLSTAT* on page 9-108
`0x0004`	`0x1`	DLCR	No	Yes	`0x00000040`	*Data Link Control Register, DLCR* on page 9-110
`0x0004`	`0x2`	TARGETID	Yes	Yes	`0x00000001`	*Target Identification Register, TARGETID* on page 9-111
`0x0004`	`0x3`	DLPIDR	Yes	Yes	`0x000000001`	*Data Link Protocol Identification Register, DLPIDR* on page 9-112
`0x0004`	`0x4`	EVENTSTAT	Yes	Yes	`0x00000001`	*Event Status Register, EVENTSTAT* on page 9-113
`0x0004`	`0x5`	SELECT1	Yes	Yes	`0x00000000`	*Select Register 1, SELECT1* on page 9-114
`0x0008`	X - on reads	RESEND	No	Yes	`0x00000000`	*Read Resend Register, RESEND* on page 9-115
`0x0008`	X - on writes	SELECT	Yes	Yes	`0x00000000`	*Select Register, SELECT* on page 9-116

Table 9-1 css600_dp register summary (continued)

Offset	DPBANKSEL	Name	JTAG-DP	SW-DP	Reset	Description
0x000C	X - on reads	RDBUFF	No	Yes	0x00000000	*Read Buffer Register, RDBUFF* on page 9-117
0x000C	X - on writes	TARGETSEL	No	Yes	0x00000000	*Target Selection Register, TARGETSEL* on page 9-118

9.2.2 Register descriptions

This section describes the `css600_dp` registers.

9.2.1 Register summary on page 9-102 provides cross references to individual registers.

AP Abort Register, ABORT

The ABORT register drives the **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 by a write to offset `0x0` of the DP register map.

The ABORT register characteristics are:

Attributes

Offset `0x0000`

Type Write-only

Reset `0x0c013477`

Width 32

The following figure shows the bit assignments.

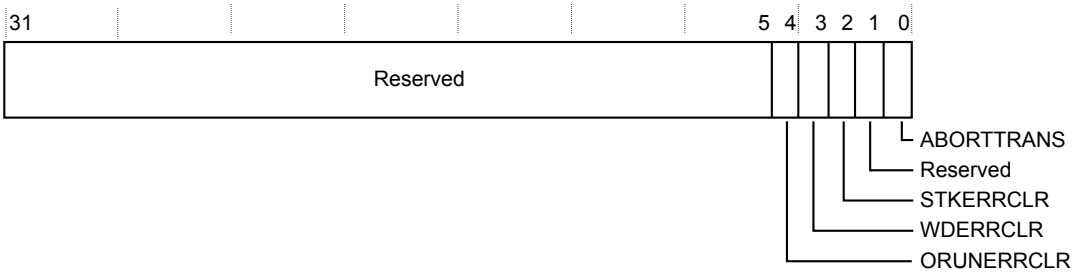

Figure 9-1 ABORT register bit assignments

The following table shows the bit assignments.

Table 9-2 ABORT register bit assignments

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

Debug Port Identification Register, DPIDR

The DPIDR provides information about the Debug Port.

The DPIDR register characteristics are:

Attributes

Offset `0x0000`
Type Read-only
Reset `0x0c013477`
Width 32

The following figure shows the bit assignments.

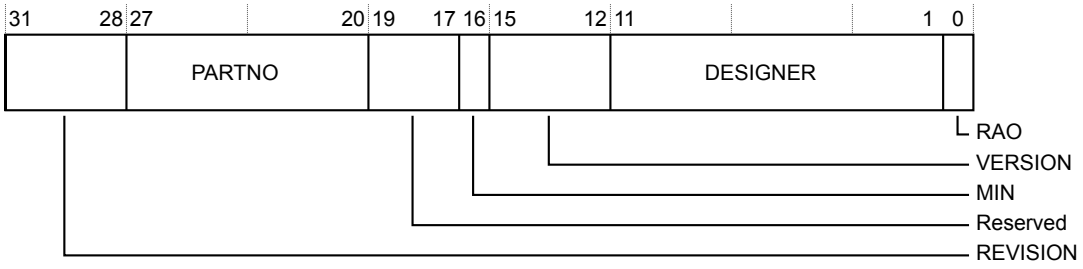

Figure 9-2 DPIDR register bit assignments

The following table shows the bit assignments.

Table 9-3 DPIDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0001`	REVISION	Revision code. `0b0001` - r1p0.
[27:20]	`0b11000000`	PARTNO	Part Number of the Debug Port.
[16]	`0b1`	MIN	Transaction counter, Pushed-verify, and Pushed-find operations are not implemented.
[15:12]	`0b0011`	VERSION	Version of Debug Port architecture implemented. SoC-600 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.

Debug Port Identification Register 1, DPIDR1

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

The DPIDR1 register characteristics are:

Attributes

Offset `0x0000`
Type Read-only
Reset `0x0000008c`
Width 32

The following figure shows the bit assignments.

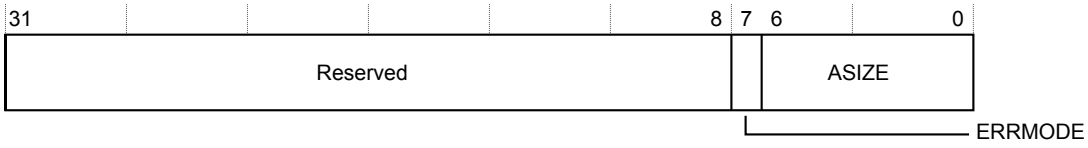

Figure 9-3 DPIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-4 DPIDR1 register bit assignments

Bits	Reset value	Name	Function
[7]	`0b1`	ERRMODE	Error reporting mode support. **0** CTRLSTAT.ERRMODE not implemented. **1** CTRLSTAT.ERRMODE implemented.
[6:0]	`0b0001100`	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.

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	`0x0000`
Type	Read-only
Reset	`0x00000032`
Width	32

The following figure shows the bit assignments.

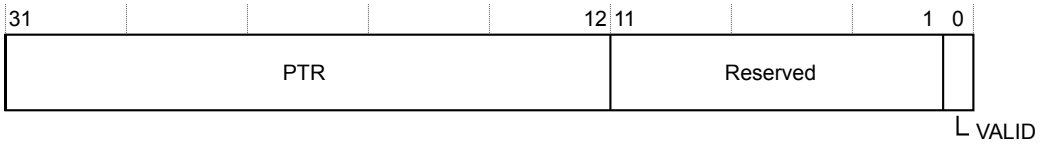

Figure 9-4 BASEPTR0 register bit assignments

The following table shows the bit assignments.

Table 9-5 BASEPTR0 register bit assignments

Bits	Reset value	Name	Function
[31:12]	`0x0`	PTR	Base address bits [31:12] of first component in the system. The address is aligned to a 4KB boundary. Depends on the interface tieoff value of **baseaddr**.
[0]	`0b0`	VALID	Indicates whether the base address is valid. Depends on the interface tieoff value of **baseaddr_valid**. **0** No base address specified. PTR is UNKNOWN. **1** Base address is specified in PTR.

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.

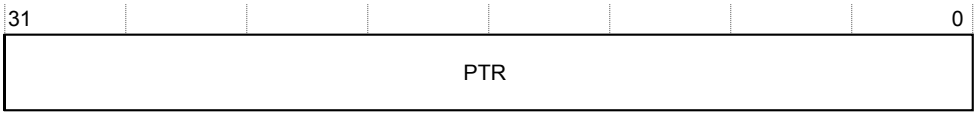

Figure 9-5 BASEPTR1 register bit assignments

The following table shows the bit assignments.

Table 9-6 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.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

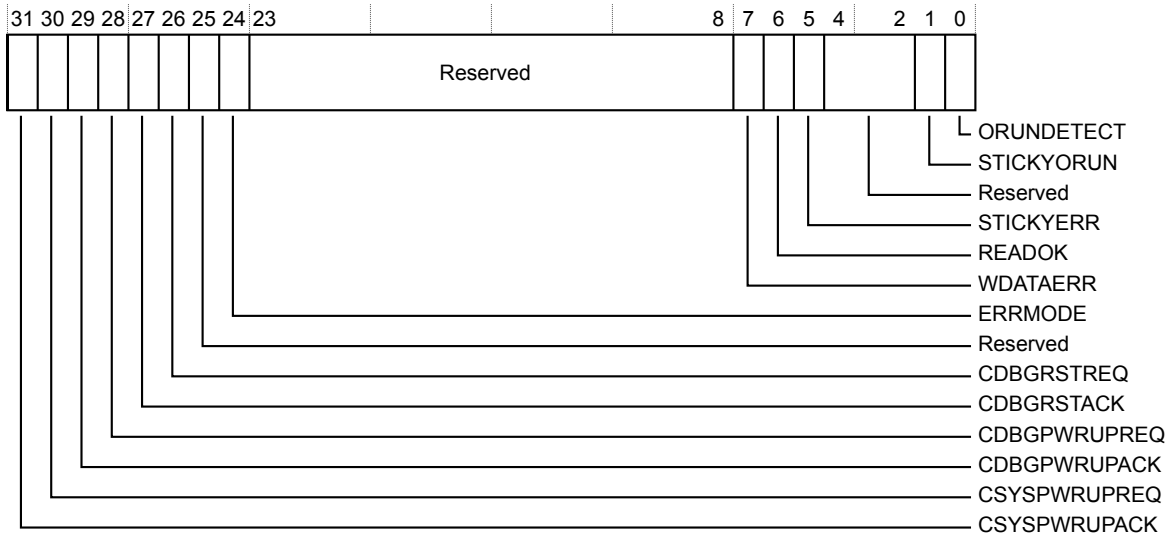

Figure 9-6 CTRLSTAT register bit assignments

The following table shows the bit assignments.

Table 9-7 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.

Table 9-7 CTRLSTAT register bit assignments (continued)

Bits	Reset value	Name	Function
[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. The bit is set to 1 if the response to the previous AP read or RDBUFF read was OK. It is cleared to 0 if the response was not OK.
[5]	0b0	STICKYERR	This bit is set to 1 if an error is returned by an AP transaction, and CTRLSTAT.ERRMODE is b0. The behavior on writing is DATA LINK DEFINED, such that on a JTAG-DP, access is R/W1C, and on a SW-DP, access is RO/WI. Clearing this bit to 0 is also DATA LINK DEFINED, such that on a JTAG-DP, writing 1 to this bit clears it, or writes 1 to the ABORT.STKERRCLR field, and on SW-DP, write 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, such that on a JTAG-DP, access is R/W1C, and on a SW-DP, access is RO/WI. Clearing this bit to 0 is also DATA LINK DEFINED, such that on a JTAG-DP, writing b1 to this bit clears it or writes a 1 to the ABORT.ORUNERRCLR field, and on SW-DP, write a 1 to the ABORT.ORUNERRCLR field.
[0]	0b0	ORUNDETECT	This bit is set to 1 to enable overrun detection.

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 programmer model 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.

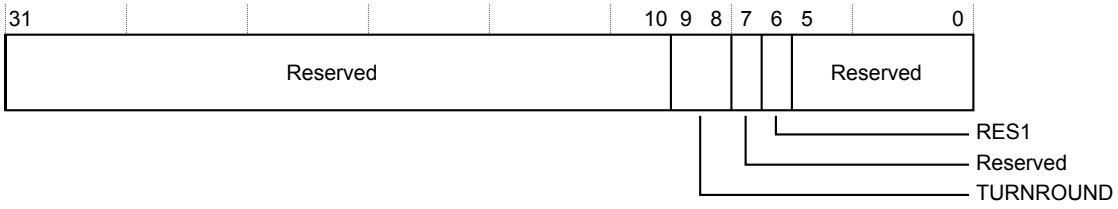

Figure 9-7 DLCR register bit assignments

The following table shows the bit assignments.

Table 9-8 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.

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	`0x00000001`
Width	32

The following figure shows the bit assignments.

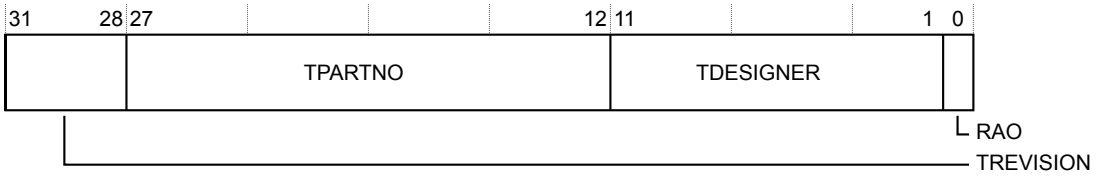

Figure 9-8 TARGETID register bit assignments

The following table shows the bit assignments.

Table 9-9 TARGETID register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0000`	TREVISION	Target revision. The value comes from the tie-off signal **targetid[31:28]**.
[27:12]	`0x0`	TPARTNO	Target part number. The value comes from the tie-off signal **targetid[27:12]**.
[11:1]	`0b00000000000`	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.

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 `0x000000001`

Width 32

The following figure shows the bit assignments.

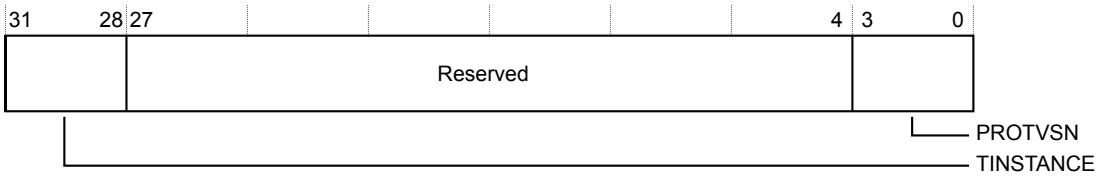

Figure 9-9 DLPIDR register bit assignments

The following table shows the bit assignments.

Table 9-10 DLPIDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0000`	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.

Event Status Register, EVENTSTAT

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

SoC-600 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 `0x00000001`
Width 32

The following figure shows the bit assignments.

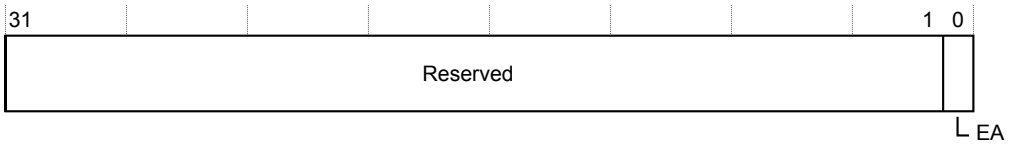

Figure 9-10 EVENTSTAT register bit assignments

The following table shows the bit assignments.

Table 9-11 EVENTSTAT register bit assignments

Bits	Reset value	Name	Function
[0]	`0b1`	EA	Event status flag. Valid values for this bit are: **0** An event requires attention. **1** There is no event requiring attention.

Select Register 1, SELECT1

The SELECT1 register is not used as CoreSight SoC-600 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.

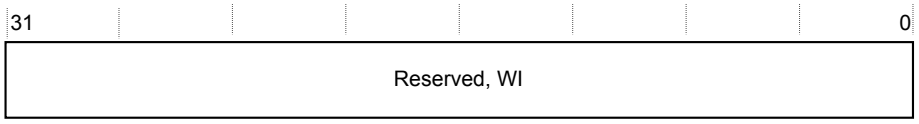

Figure 9-11 SELECT1 register bit assignments

The following table shows the bit assignments.

Table 9-12 SELECT1 register bit assignments

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

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.

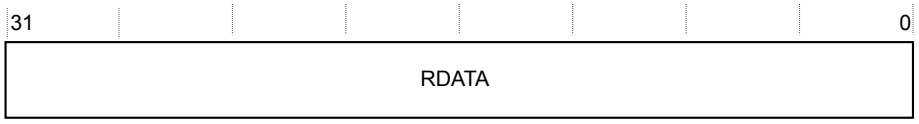

Figure 9-12 RESEND register bit assignments

The following table shows the bit assignments.

Table 9-13 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.

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.

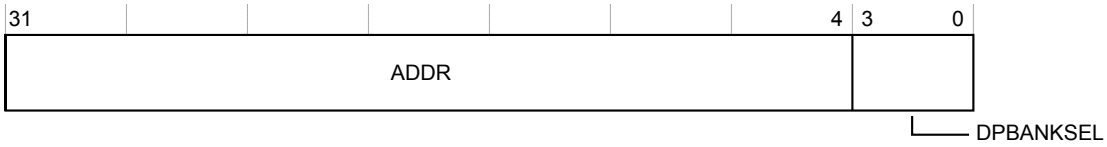

Figure 9-13 SELECT register bit assignments

The following table shows the bit assignments.

Table 9-14 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.

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.

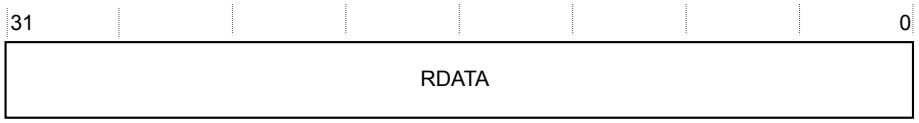

Figure 9-14 RDBUFF register bit assignments

The following table shows the bit assignments.

Table 9-15 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 last AP read access.

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 programmer model 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.

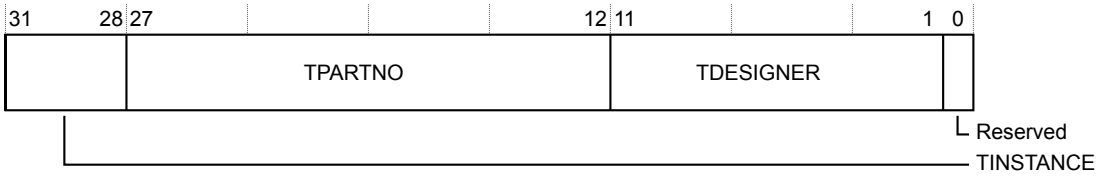

Figure 9-15 TARGETSEL register bit assignments

The following table shows the bit assignments.

Table 9-16 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.

9.3 css600_apbap introduction

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

This section contains the following subsections:

- *9.3.1 Register summary* on page 9-119.
- *9.3.2 Register descriptions* on page 9-121.

9.3.1 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`.

Table 9-17 css600_apbap - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
`0x0000`	DAR0	RW	`0x--------`	32	*Direct Access Register 0, DAR0* on page 9-122
`0x0004`	DAR1	RW	`0x--------`	32	*Direct Access Register 1, DAR1* on page 9-123
`0x0008`	DAR2	RW	`0x--------`	32	*Direct Access Register 2, DAR2* on page 9-124
…	…	…	…	…	…
`0x03FC`	DAR255	RW	`0x--------`	32	*Direct Access Register 255, DAR255* on page 9-125
`0x0D00`	CSW	RW	`0x30-000-2`	32	*Control Status Word register, CSW* on page 9-126
`0x0D04`	TAR	RW	`0x--------`	32	*Transfer Address Register, TAR* on page 9-128
`0x0D0C`	DRW	RW	`0x--------`	32	*Data Read/Write register, DRW* on page 9-129
`0x0D10`	BD0	RW	`0x--------`	32	*Banked Data register 0, BD0* on page 9-130
`0x0D14`	BD1	RW	`0x--------`	32	*Banked Data register 1, BD1* on page 9-131
`0x0D18`	BD2	RW	`0x--------`	32	*Banked Data register 2, BD2* on page 9-132
`0x0D1C`	BD3	RW	`0x--------`	32	*Banked Data register 3, BD3* on page 9-133
`0x0D24`	TRR	RW	`0x00000000`	32	*Transfer Response Register, TRR* on page 9-134
`0x0DF4`	CFG	RO	`0x000101a0`	32	*Configuration register, CFG* on page 9-135
`0x0DF8`	BASE	RO	`0x-----00-`	32	*Debug Base Address register, BASE* on page 9-136
`0x0DFC`	IDR	RO	`0x04770006`	32	*Identification Register, IDR* on page 9-137
`0x0EFC`	ITSTATUS	RW	`0x00000000`	32	*Integration Test Status register, ITSTATUS* on page 9-138
`0x0F00`	ITCTRL	RW	`0x00000000`	32	*Integration Mode Control Register, ITCTRL* on page 9-139
`0x0FA0`	CLAIMSET	RW	`0x00000003`	32	*Claim Tag Set Register, CLAIMSET* on page 9-140
`0x0FA4`	CLAIMCLR	RW	`0x00000000`	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-141
`0x0FB8`	AUTHSTATUS	RO	`0x000000--`	32	*Authentication Status Register, AUTHSTATUS* on page 9-142

Table 9-17 css600_apbap - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FBC	DEVARCH	RO	0x47700a17	32	*Device Architecture Register, DEVARCH* on page 9-144
0x0FCC	DEVTYPE	RO	0x00000000	32	*Device Type Identifier Register, DEVTYPE* on page 9-145
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-146
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-147
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-148
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-149
0x0FE0	PIDR0	RO	0x000000e2	32	*Peripheral Identification Register 0, PIDR0* on page 9-150
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-151
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-152
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-153
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-154
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-155
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-156
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-157
0x1000	DAR0	RW	0x--------	32	*Direct Access Register 0, DAR0* on page 9-122
0x1004	DAR1	RW	0x--------	32	*Direct Access Register 1, DAR1* on page 9-123
0x1008	DAR2	RW	0x--------	32	*Direct Access Register 2, DAR2* on page 9-124
…	…	…	…	…	…
0x13FC	DAR255	RW	0x--------	32	*Direct Access Register 255, DAR255* on page 9-125
0x1D00	CSW	RW	0x30-000-2	32	*Control Status Word register, CSW* on page 9-126
0x1D04	TAR	RW	0x--------	32	*Transfer Address Register, TAR* on page 9-128
0x1D0C	DRW	RW	0x--------	32	*Data Read/Write register, DRW* on page 9-129
0x1D10	BD0	RW	0x--------	32	*Banked Data register 0, BD0* on page 9-130
0x1D14	BD1	RW	0x--------	32	*Banked Data register 1, BD1* on page 9-131
0x1D18	BD2	RW	0x--------	32	*Banked Data register 2, BD2* on page 9-132
0x1D1C	BD3	RW	0x--------	32	*Banked Data register 3, BD3* on page 9-133
0x1D24	TRR	RW	0x00000000	32	*Transfer Response Register, TRR* on page 9-134
0x1DF4	CFG	RO	0x000101a0	32	*Configuration register, CFG* on page 9-135
0x1DF8	BASE	RO	0x-----00-	32	*Debug Base Address register, BASE* on page 9-136
0x1DFC	IDR	RO	0x04770006	32	*Identification Register, IDR* on page 9-137
0x1EFC	ITSTATUS	RW	0x00000000	32	*Integration Test Status register, ITSTATUS* on page 9-138
0x1F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-139
0x1FA0	CLAIMSET	RW	0x00000003	32	*Claim Tag Set Register, CLAIMSET* on page 9-140
0x1FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-141
0x1FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-142

Table 9-17 css600_apbap - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x1FBC	DEVARCH	RO	0x47700a17	32	*Device Architecture Register, DEVARCH* on page 9-144
0x1FCC	DEVTYPE	RO	0x00000000	32	*Device Type Identifier Register, DEVTYPE* on page 9-145
0x1FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-146
0x1FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-147
0x1FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-148
0x1FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-149
0x1FE0	PIDR0	RO	0x000000e2	32	*Peripheral Identification Register 0, PIDR0* on page 9-150
0x1FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-151
0x1FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-152
0x1FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-153
0x1FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-154
0x1FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-155
0x1FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-156
0x1FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-157

9.3.2 Register descriptions

This section describes the `css600_apbap` registers.

9.3.1 Register summary on page 9-119 provides cross references to individual registers.

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.

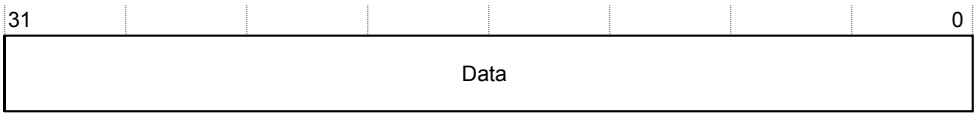

Figure 9-16 DAR0 register bit assignments

The following table shows the bit assignments.

Table 9-18 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.

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.

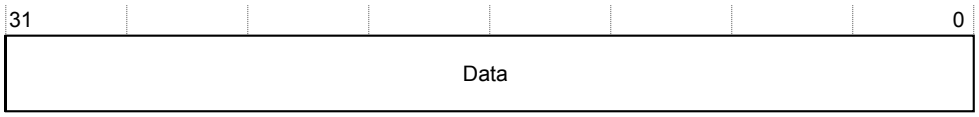

Figure 9-17 DAR1 register bit assignments

The following table shows the bit assignments.

Table 9-19 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.

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.

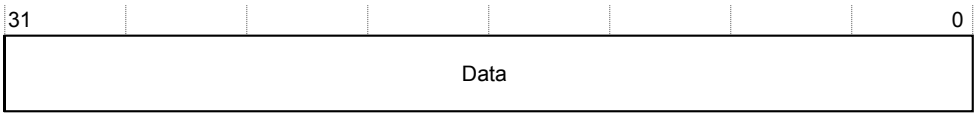

Figure 9-18 DAR2 register bit assignments

The following table shows the bit assignments.

Table 9-20 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.

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.

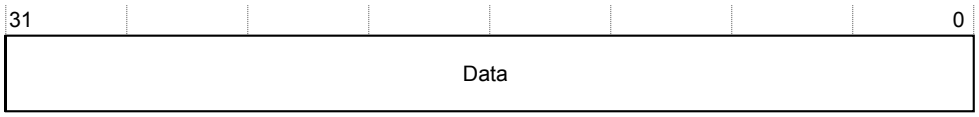

Figure 9-19 DAR255 register bit assignments

The following table shows the bit assignments.

Table 9-21 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.

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.

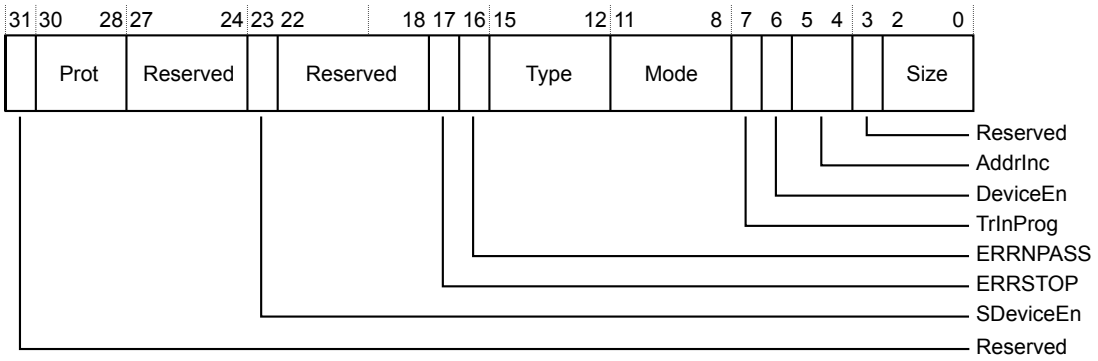

Figure 9-20 CSW register bit assignments

The following table shows the bit assignments.

Table 9-22 CSW register bit assignments

Bits	Reset value	Name	Function
[30:28]	`0b011`	Prot	Drives APB master interface port **pprot_m[2:0]** which specifies the APB4 protection encoding. The reset value is `0x3` (Data, Non-secure, Privileged). Together with authentication interface signals, CSW.Prot[1] determines whether a Secure access is allowed on the master interface as follows, access = **dbgen** && **spiden** || **dbgen** && CSW.Prot[1].
[23]	UNKNOWN	SDeviceEn	Indicates the status of the **spiden** and **spniden** ports. It is set when either **spiden** or **spniden** is HIGH, and remains clear otherwise. If this bit is clear, Secure APB master 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.
[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 that are not passed upstream. Reset to 0. **0** Memory access errors that are passed upstream. **1** Memory access errors that 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.

Table 9-22 CSW register bit assignments (continued)

Bits	Reset value	Name	Function
[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 **dbgen** and **niden** ports. The bit is set when either **dbgen** or **niden** is HIGH, and is clear otherwise. If this bit is clear, no APB master 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.
[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.

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.

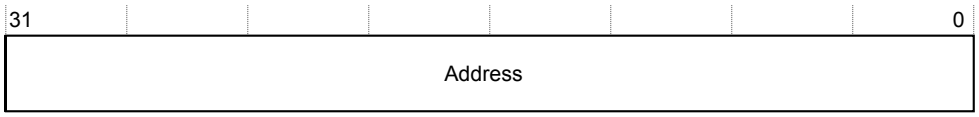

Figure 9-21 TAR register bit assignments

The following table shows the bit assignments.

Table 9-23 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.

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.

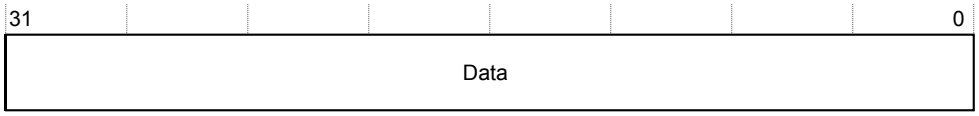

Figure 9-22 DRW register bit assignments

The following table shows the bit assignments.

Table 9-24 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.

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.

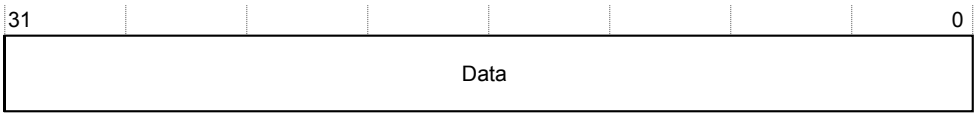

Figure 9-23 BD0 register bit assignments

The following table shows the bit assignments.

Table 9-25 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.

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.

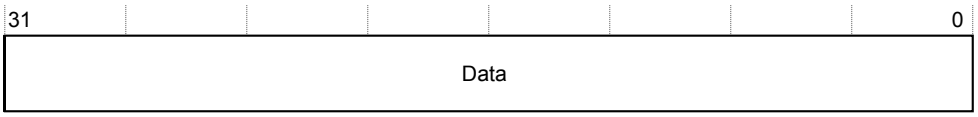

Figure 9-24 BD1 register bit assignments

The following table shows the bit assignments.

Table 9-26 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.

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.

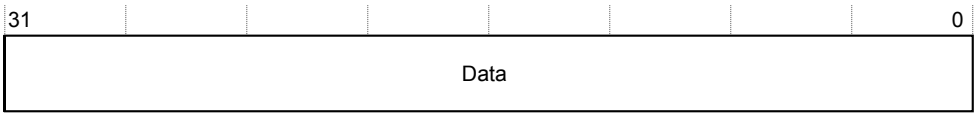

Figure 9-25 BD2 register bit assignments

The following table shows the bit assignments.

Table 9-27 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.

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.

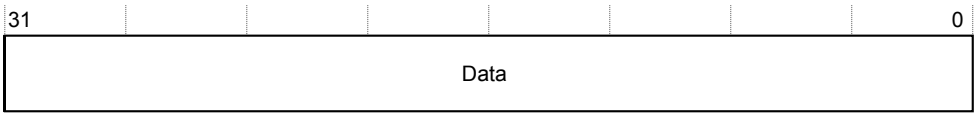

Figure 9-26 BD3 register bit assignments

The following table shows the bit assignments.

Table 9-28 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.

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.

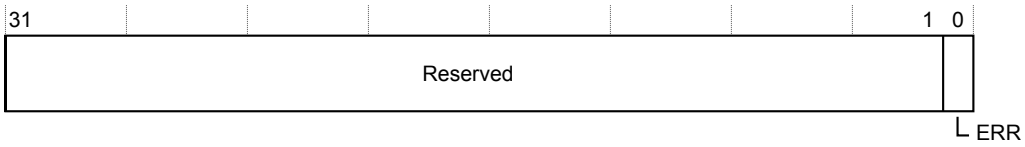

Figure 9-27 TRR register bit assignments

The following table shows the bit assignments.

Table 9-29 TRR register bit assignments

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.

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.

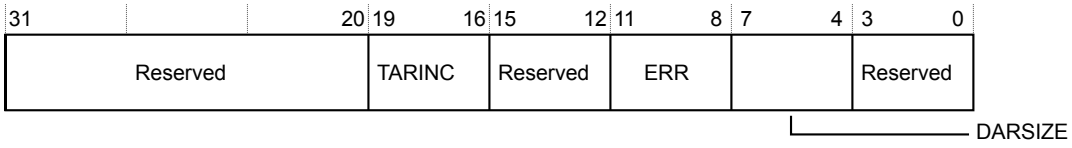

Figure 9-28 CFG register bit assignments

The following table shows the bit assignments.

Table 9-30 CFG register bit assignments

Bits	Reset value	Name	Function
[19:16]	`0b0001`	TARINC	TAR incrementer size. Returns `0x1` indicating a TAR incrementer size of 10-bits.
[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.

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.

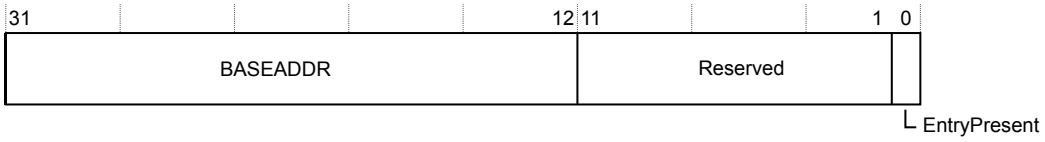

Figure 9-29 BASE register bit assignments

The following table shows the bit assignments.

Table 9-31 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 port **baseaddr[31:12]**, otherwise, it reads as `0x0`.
[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 port **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.

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 `0x04770006`
Width 32

The following figure shows the bit assignments.

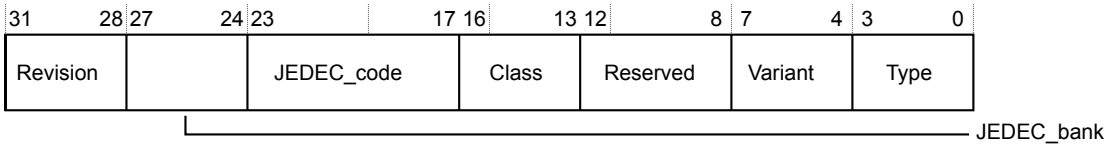

Figure 9-30 IDR register bit assignments

The following table shows the bit assignments.

Table 9-32 IDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0000`	Revision	Returns `0x0` (r0p0).
[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.

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.

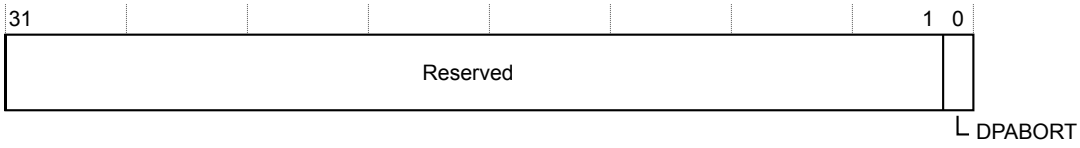

Figure 9-31 ITSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-33 ITSTATUS register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	DPABORT	When in Integration testing mode (f.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.

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.

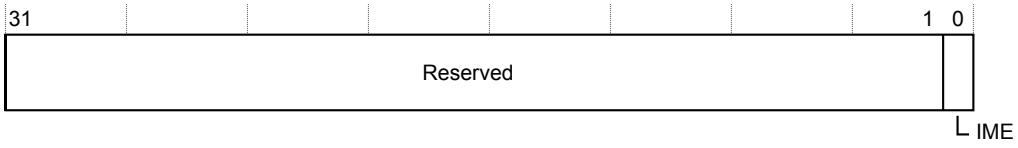

Figure 9-32 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-34 ITCTRL register bit assignments

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

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.

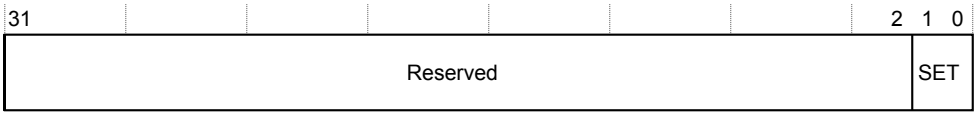

Figure 9-33 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-35 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[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.

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.

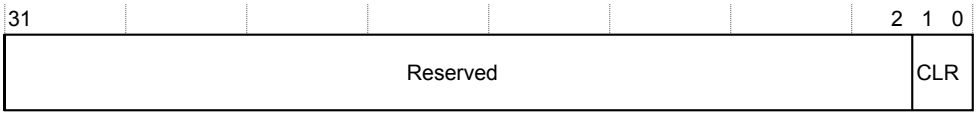

Figure 9-34 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-36 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[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.

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.

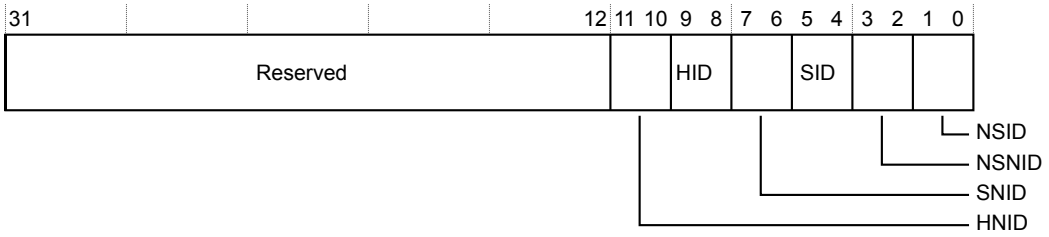

Figure 9-35 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-37 AUTHSTATUS register bit assignments

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.

Table 9-37 AUTHSTATUS register bit assignments (continued)

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.

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.

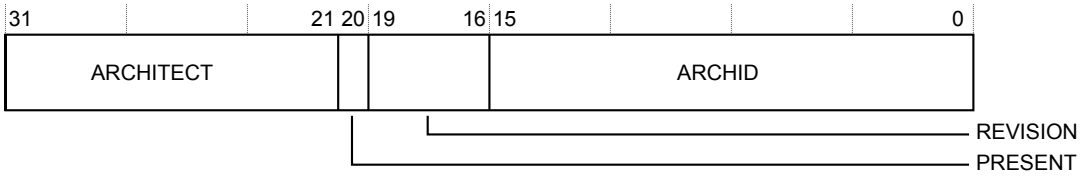

Figure 9-36 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-38 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.

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.

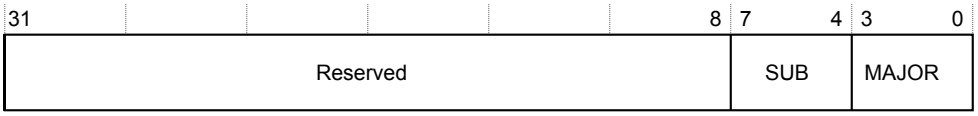

Figure 9-37 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-39 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	SUB	Minor classification. Returns `0x0`, Other/undefined.
[3:0]	`0b0000`	MAJOR	Major classification. Returns `0x0`, Miscellaneous.

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.

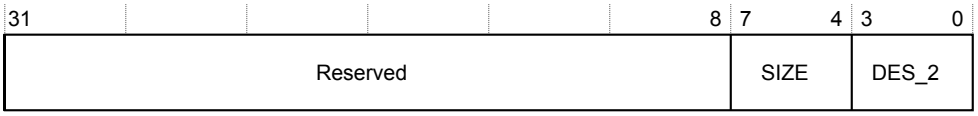

Figure 9-38 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-40 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

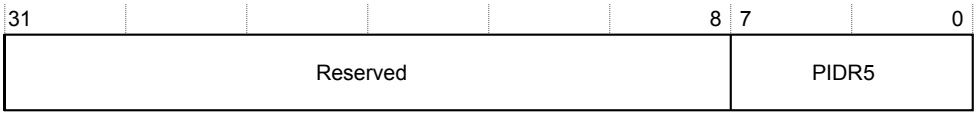

Figure 9-39 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-41 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

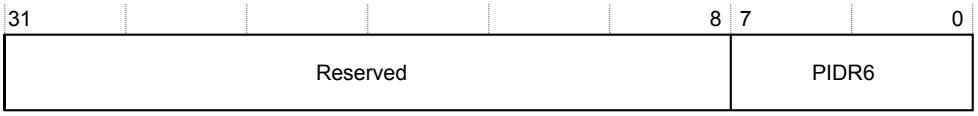

Figure 9-40 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-42 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

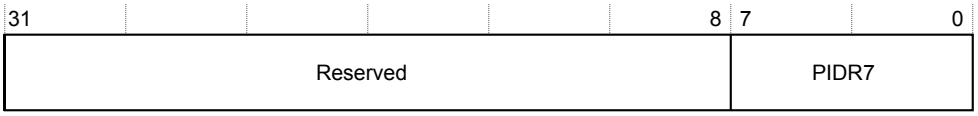

Figure 9-41 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-43 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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.

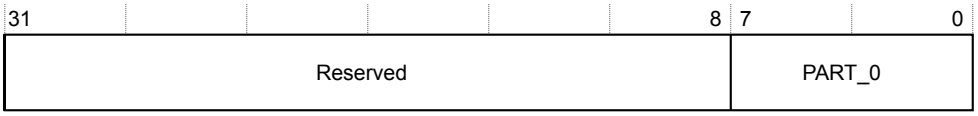

Figure 9-42 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-44 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11100010`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

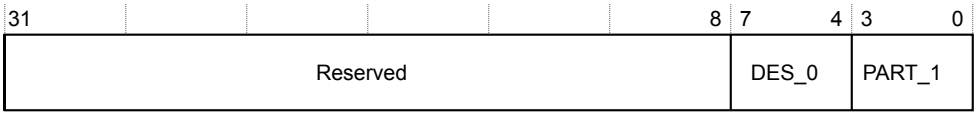

Figure 9-43 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-45 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x0000000b`
Width 32

The following figure shows the bit assignments.

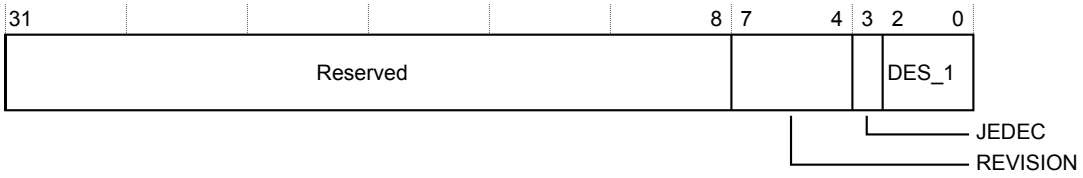

Figure 9-44 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-46 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

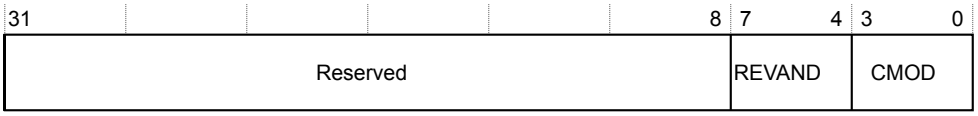

Figure 9-45 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-47 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

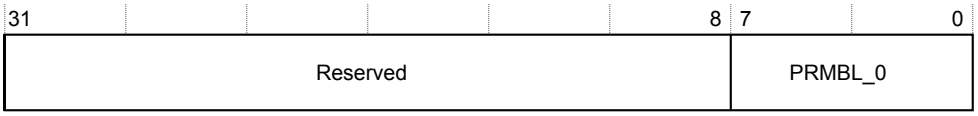

Figure 9-46 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-48 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

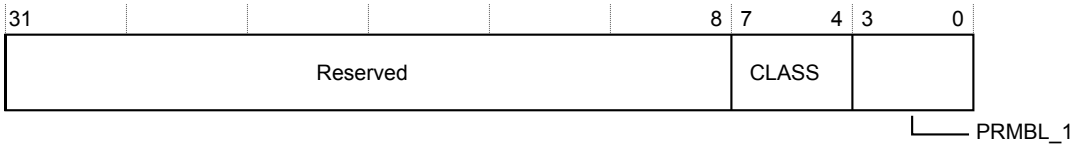

Figure 9-47 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-49 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

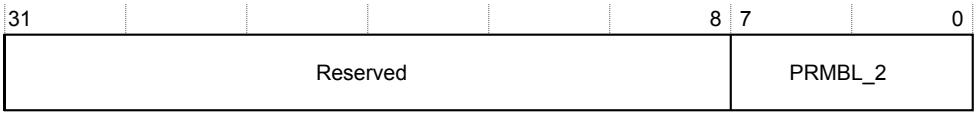

Figure 9-48 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-50 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

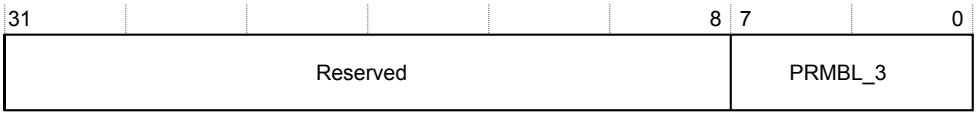

Figure 9-49 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-51 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.4 css600_ahbap introduction

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

This section contains the following subsections:

- *9.4.1 Register summary* on page 9-158.
- *9.4.2 Register descriptions* on page 9-160.

9.4.1 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`.

Table 9-52 css600_ahbap - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	DAR0	RW	0x--------	32	*Direct Access Register 0, DAR0* on page 9-161
0x0004	DAR1	RW	0x--------	32	*Direct Access Register 1, DAR1* on page 9-162
0x0008	DAR2	RW	0x--------	32	*Direct Access Register 2, DAR2* on page 9-163
…	…	…	…	…	…
0x03FC	DAR255	RW	0x--------	32	*Direct Access Register 255, DAR255* on page 9-164
0x0D00	CSW	RW	0x43-000-2	32	*Control Status Word register, CSW* on page 9-165
0x0D04	TAR	RW	0x--------	32	*Transfer Address Register, TAR* on page 9-167
0x0D0C	DRW	RW	0x--------	32	*Data Read/Write register, DRW* on page 9-168
0x0D10	BD0	RW	0x--------	32	*Banked Data register 0, BD0* on page 9-169
0x0D14	BD1	RW	0x--------	32	*Banked Data register 1, BD1* on page 9-170
0x0D18	BD2	RW	0x--------	32	*Banked Data register 2, BD2* on page 9-171
0x0D1C	BD3	RW	0x--------	32	*Banked Data register 3, BD3* on page 9-172
0x0D24	TRR	RW	0x00000000	32	*Transfer Response Register, TRR* on page 9-173
0x0DF4	CFG	RO	0x000101a0	32	*Configuration register, CFG* on page 9-174
0x0DF8	BASE	RO	0x-----00-	32	*Debug Base Address register, BASE* on page 9-175
0x0DFC	IDR	RO	0x04770005	32	*Identification Register, IDR* on page 9-176
0x0EFC	ITSTATUS	RW	0x00000000	32	*Integration Test Status register, ITSTATUS* on page 9-177
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-178
0x0FA0	CLAIMSET	RW	0x00000003	32	*Claim Tag Set Register, CLAIMSET* on page 9-179
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-180
0x0FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-181

Table 9-52 css600_ahbap - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FBC	DEVARCH	RO	0x47700a17	32	*Device Architecture Register, DEVARCH* on page 9-183
0x0FCC	DEVTYPE	RO	0x00000000	32	*Device Type Identifier Register, DEVTYPE* on page 9-184
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-185
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-186
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-187
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-188
0x0FE0	PIDR0	RO	0x000000e3	32	*Peripheral Identification Register 0, PIDR0* on page 9-189
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-190
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-191
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-192
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-193
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-194
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-195
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-196
0x1000	DAR0	RW	0x--------	32	*Direct Access Register 0, DAR0* on page 9-161
0x1004	DAR1	RW	0x--------	32	*Direct Access Register 1, DAR1* on page 9-162
0x1008	DAR2	RW	0x--------	32	*Direct Access Register 2, DAR2* on page 9-163
…	…	…	…	…	…
0x13FC	DAR255	RW	0x--------	32	*Direct Access Register 255, DAR255* on page 9-164
0x1D00	CSW	RW	0x43-000-2	32	*Control Status Word register, CSW* on page 9-165
0x1D04	TAR	RW	0x--------	32	*Transfer Address Register, TAR* on page 9-167
0x1D0C	DRW	RW	0x--------	32	*Data Read/Write register, DRW* on page 9-168
0x1D10	BD0	RW	0x--------	32	*Banked Data register 0, BD0* on page 9-169
0x1D14	BD1	RW	0x--------	32	*Banked Data register 1, BD1* on page 9-170
0x1D18	BD2	RW	0x--------	32	*Banked Data register 2, BD2* on page 9-171
0x1D1C	BD3	RW	0x--------	32	*Banked Data register 3, BD3* on page 9-172
0x1D24	TRR	RW	0x00000000	32	*Transfer Response Register, TRR* on page 9-173
0x1DF4	CFG	RO	0x000101a0	32	*Configuration register, CFG* on page 9-174
0x1DF8	BASE	RO	0x-----00-	32	*Debug Base Address register, BASE* on page 9-175
0x1DFC	IDR	RO	0x04770005	32	*Identification Register, IDR* on page 9-176
0x1EFC	ITSTATUS	RW	0x00000000	32	*Integration Test Status register, ITSTATUS* on page 9-177
0x1F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-178
0x1FA0	CLAIMSET	RW	0x00000003	32	*Claim Tag Set Register, CLAIMSET* on page 9-179
0x1FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-180
0x1FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-181

Table 9-52 css600_ahbap - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x1FBC	DEVARCH	RO	0x47700a17	32	*Device Architecture Register, DEVARCH* on page 9-183
0x1FCC	DEVTYPE	RO	0x00000000	32	*Device Type Identifier Register, DEVTYPE* on page 9-184
0x1FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-185
0x1FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-186
0x1FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-187
0x1FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-188
0x1FE0	PIDR0	RO	0x000000e3	32	*Peripheral Identification Register 0, PIDR0* on page 9-189
0x1FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-190
0x1FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-191
0x1FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-192
0x1FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-193
0x1FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-194
0x1FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-195
0x1FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-196

9.4.2 Register descriptions

This section describes the `css600_ahbap` registers.

9.4.1 Register summary on page 9-158 provides cross references to individual registers.

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.

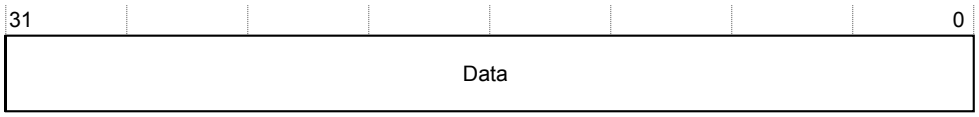

Figure 9-50 DAR0 register bit assignments

The following table shows the bit assignments.

Table 9-53 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.

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.

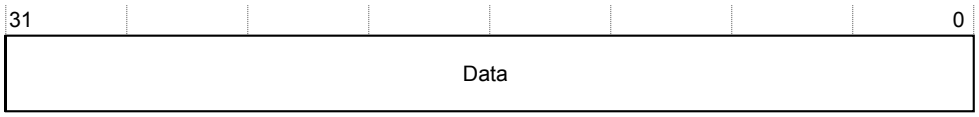

Figure 9-51 DAR1 register bit assignments

The following table shows the bit assignments.

Table 9-54 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.

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.

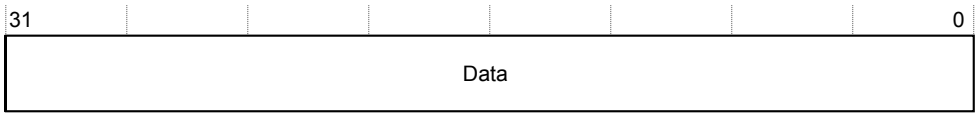

Figure 9-52 DAR2 register bit assignments

The following table shows the bit assignments.

Table 9-55 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.

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.

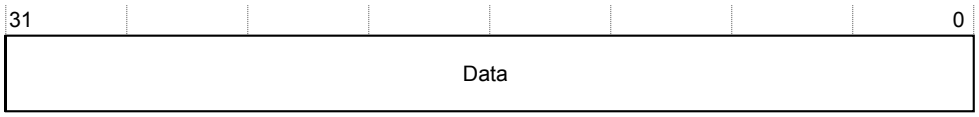

Figure 9-53 DAR255 register bit assignments

The following table shows the bit assignments.

Table 9-56 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.

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.

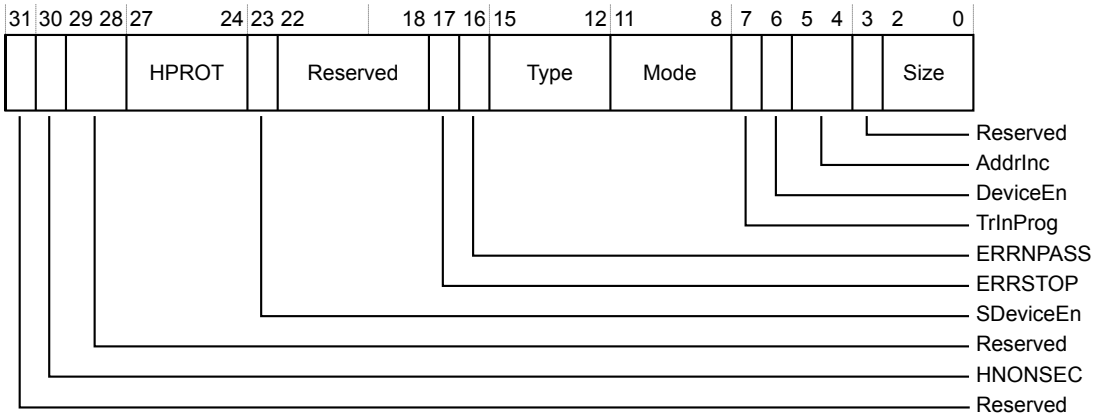

Figure 9-54 CSW register bit assignments

The following table shows the bit assignments.

Table 9-57 CSW register bit assignments

Bits	Reset value	Name	Function
[30]	`0b1`	HNONSEC	Drives **hnonsec_m** output pin. Together with authentication interface signals HNONSEC determines whether a secure access is allowed on the master interface as follows, access = **dbgen** && **spiden** || **dbgen** && HNONSEC.
[27:24]	`0b0011`	HPROT	Specifies the protection signal encoding to be output on **hprot_m[3:0]**. CSW.HPROT[3] also drives **hprot_m[6]** and **hprot_m[4]**. Reset to `0x3` (Non-Shareable, Non-Lookup, Non-Modifiable, Non-Bufferable, Privileged, Data).
[23]	UNKNOWN	SDeviceEn	Indicates the status of the **spiden** and **spniden** ports. It is set when either **spiden** or **spniden** is 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.
[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 that are not passed upstream. **0** Memory access errors that are passed upstream. **1** Memory access errors that are not passed upstream.

Table 9-57 CSW register bit assignments (continued)

Bits	Reset value	Name	Function
[15:12]	0b0000	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 port. If the master interface is busy, CSW.TrInProg is set in both logical APs.
[6]	UNKNOWN	DeviceEn	Indicates the status of **dbgen** and **niden** ports. The bit is set when either **dbgen** or **niden** 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.
[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.

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.

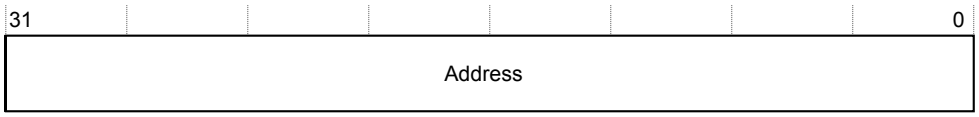

Figure 9-55 TAR register bit assignments

The following table shows the bit assignments.

Table 9-58 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.

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.

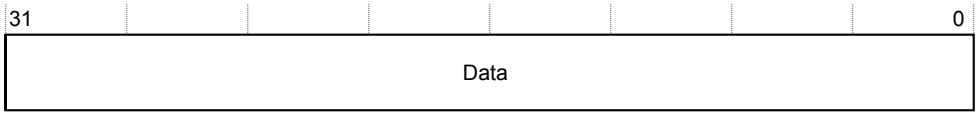

Figure 9-56 DRW register bit assignments

The following table shows the bit assignments.

Table 9-59 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.

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.

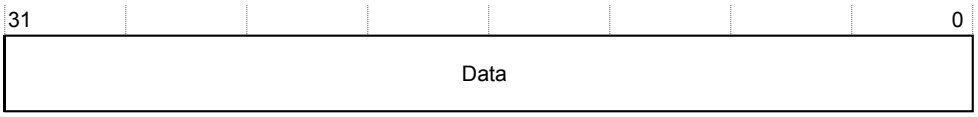

Figure 9-57 BD0 register bit assignments

The following table shows the bit assignments.

Table 9-60 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.

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.

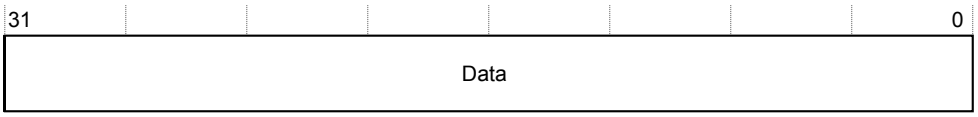

Figure 9-58 BD1 register bit assignments

The following table shows the bit assignments.

Table 9-61 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.

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.

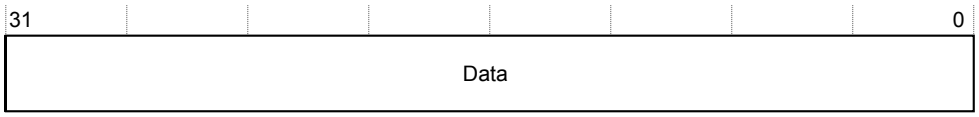

Figure 9-59 BD2 register bit assignments

The following table shows the bit assignments.

Table 9-62 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.

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.

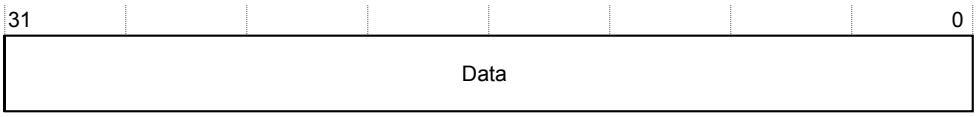

Figure 9-60 BD3 register bit assignments

The following table shows the bit assignments.

Table 9-63 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.

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.

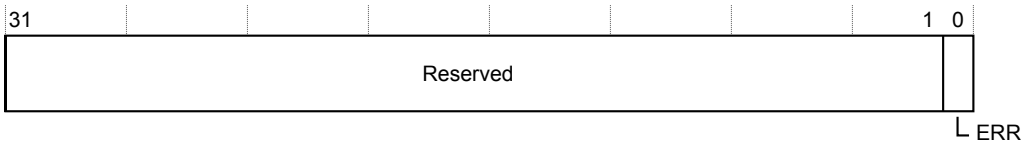

Figure 9-61 TRR register bit assignments

The following table shows the bit assignments.

Table 9-64 TRR register bit assignments

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.

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.

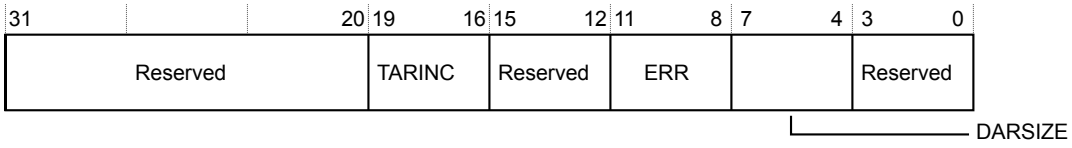

Figure 9-62 CFG register bit assignments

The following table shows the bit assignments.

Table 9-65 CFG register bit assignments

Bits	Reset value	Name	Function
[19:16]	`0b0001`	TARINC	TAR incrementer size. Returns `0x1` indicating a TAR incrementer size of 10-bits.
[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.

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.

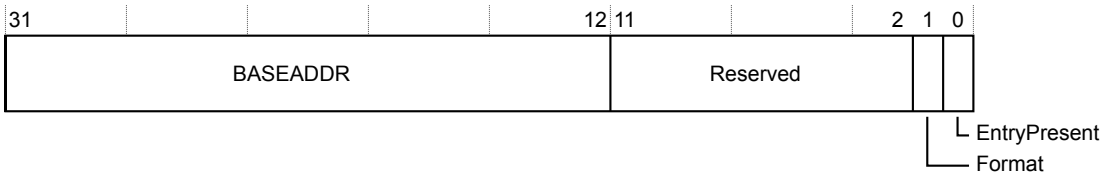

Figure 9-63 BASE register bit assignments

The following table shows the bit assignments.

Table 9-66 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 port **baseaddr[31:12]**, otherwise, it reads as `0x0`.
[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 port **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.

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 0x04770005
Width 32

The following figure shows the bit assignments.

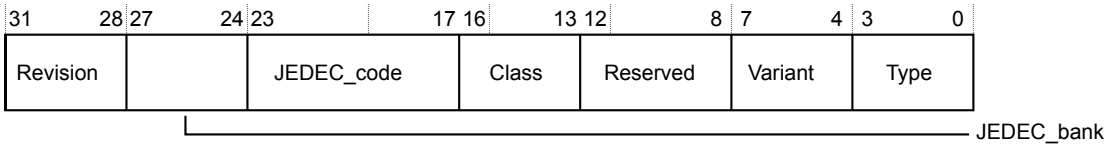

Figure 9-64 IDR register bit assignments

The following table shows the bit assignments.

Table 9-67 IDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	0b0000	Revision	Returns 0x0 (r0p0).
[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.
[7:4]	0b0000	Variant	Returns 0x0, indicating no variation from base type specified by IDR.Type.
[3:0]	0b0101	Type	Returns 0x5, indicating that this is an AHB5 Access Port.

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.

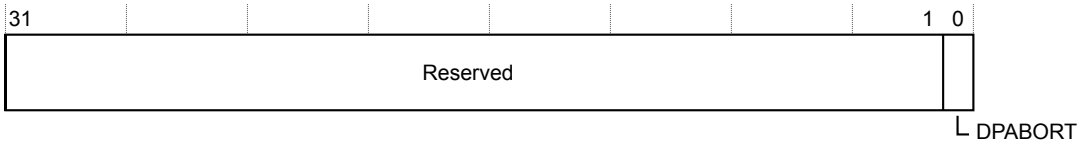

Figure 9-65 ITSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-68 ITSTATUS register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	DPABORT	When in Integration testing mode (f.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.

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.

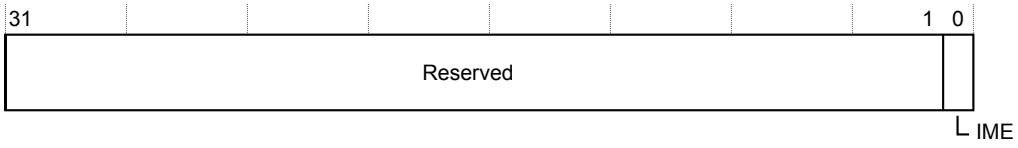

Figure 9-66 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-69 ITCTRL register bit assignments

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

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.

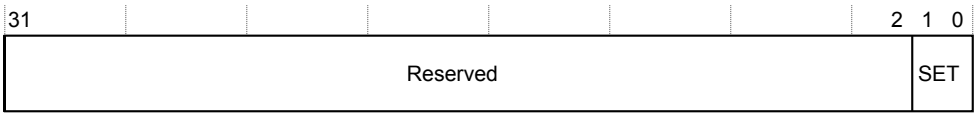

Figure 9-67 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-70 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[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.

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.

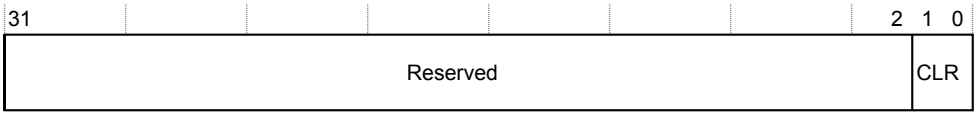

Figure 9-68 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-71 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[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.

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.

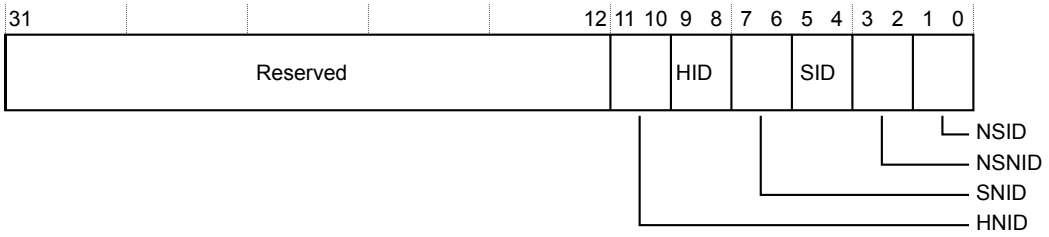

Figure 9-69 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-72 AUTHSTATUS register bit assignments

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.

Table 9-72 AUTHSTATUS register bit assignments (continued)

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.

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.

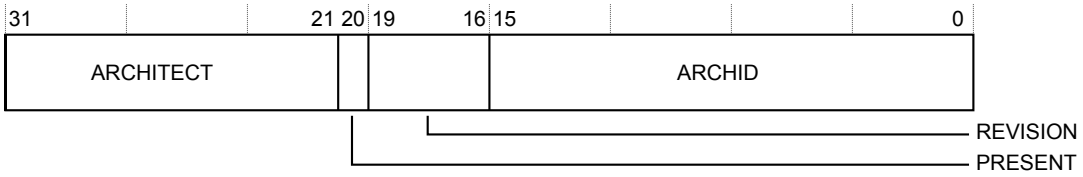

Figure 9-70 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-73 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.

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.

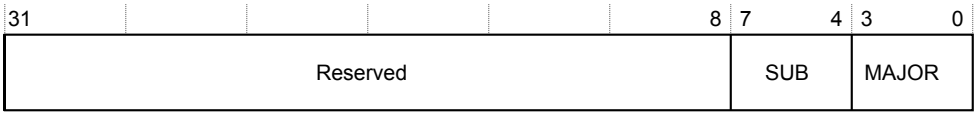

Figure 9-71 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-74 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	SUB	Minor classification. Returns `0x0`, Other/undefined.
[3:0]	`0b0000`	MAJOR	Major classification. Returns `0x0`, Miscellaneous.

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.

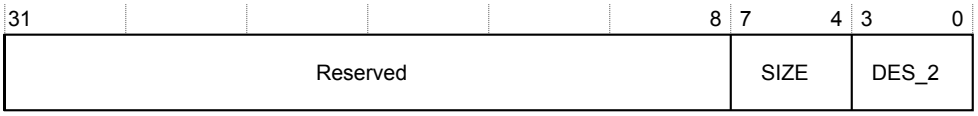

Figure 9-72 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-75 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

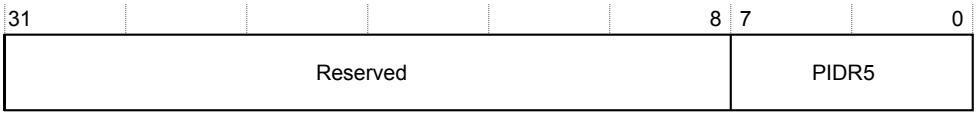

Figure 9-73 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-76 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	0b00000000	PIDR5	Reserved.

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.

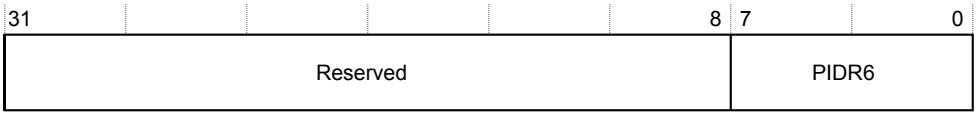

Figure 9-74 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-77 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

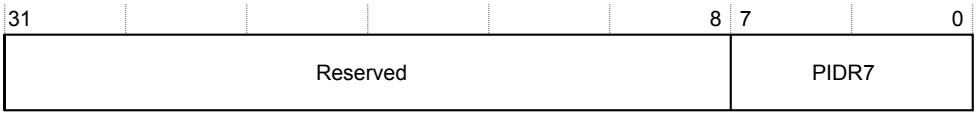

Figure 9-75 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-78 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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.

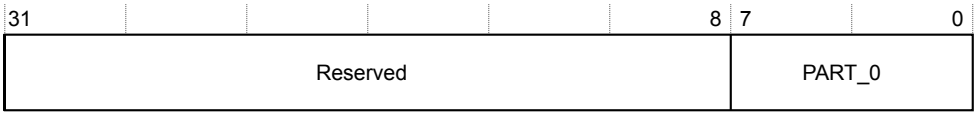

Figure 9-76 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-79 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11100011`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

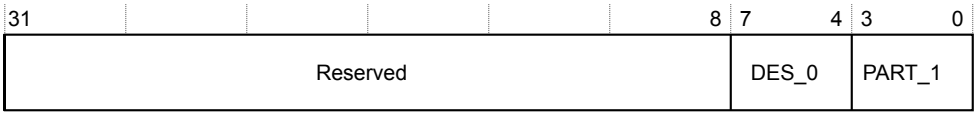

Figure 9-77 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-80 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x0000000b`
Width 32

The following figure shows the bit assignments.

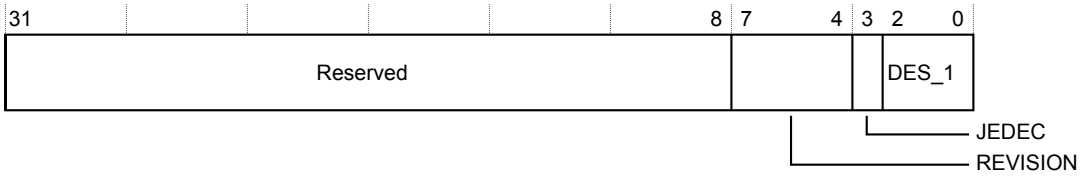

Figure 9-78 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-81 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

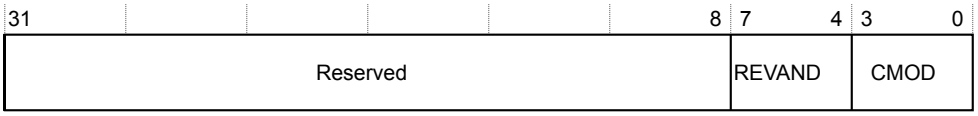

Figure 9-79 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-82 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

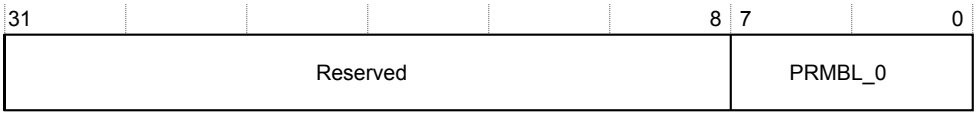

Figure 9-80 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-83 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

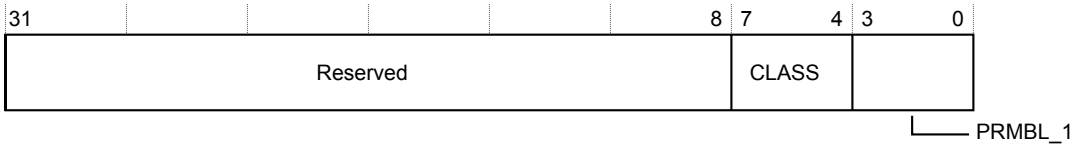

Figure 9-81 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-84 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

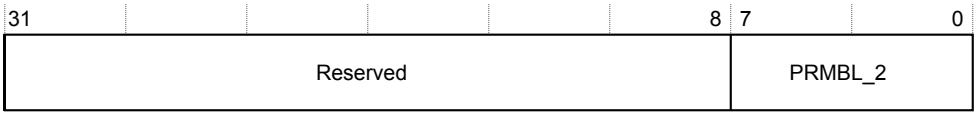

Figure 9-82 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-85 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

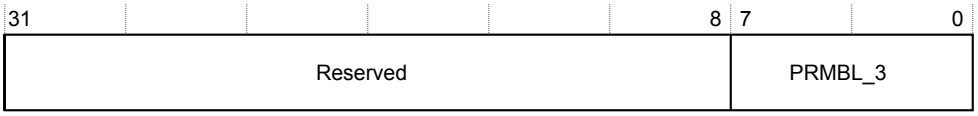

Figure 9-83 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-86 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.5 css600_axiap introduction

This section describes the functions and programmers model of the `css600_axiap`.

This section contains the following subsections:

- *9.5.1 Register summary* on page 9-197.
- *9.5.2 Register descriptions* on page 9-199.

9.5.1 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`.

Table 9-87 css600_axiap - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	DAR0	RW	0x--------	32	*Direct Access Register 0, DAR0* on page 9-200
0x0004	DAR1	RW	0x--------	32	*Direct Access Register 1, DAR1* on page 9-201
0x0008	DAR2	RW	0x--------	32	*Direct Access Register 2, DAR2* on page 9-202
…	…	…	…	…	…
0x03FC	DAR255	RW	0x--------	32	*Direct Access Register 255, DAR255* on page 9-203
0x0D00	CSW	RW	0x30-060-2	32	*Control Status Word register, CSW* on page 9-204
0x0D04	TAR	RW	0x--------	32	*Transfer Address Register, TAR* on page 9-206
0x0D08	TARH	RW	0x00000000	32	*Transfer Address Register, TARH* on page 9-207
0x0D0C	DRW	RW	0x--------	32	*Data Read/Write register, DRW* on page 9-208
0x0D10	BD0	RW	0x--------	32	*Banked Data register 0, BD0* on page 9-209
0x0D14	BD1	RW	0x--------	32	*Banked Data register 1, BD1* on page 9-210
0x0D18	BD2	RW	0x--------	32	*Banked Data register 2, BD2* on page 9-211
0x0D1C	BD3	RW	0x--------	32	*Banked Data register 3, BD3* on page 9-212
0x0D20	MBT	RW	0x00000000	32	*Memory Barrier Transfer register, MBT* on page 9-213
0x0D24	TRR	RW	0x00000000	32	*Transfer Response Register, TRR* on page 9-214
0x0DF0	BASEH	RW	0x--------	32	*Debug Base Address register upper 32 bits, BASEH* on page 9-215
0x0DF4	CFG	RO	0x000101a0	32	*Configuration register, CFG* on page 9-216
0x0DF8	BASE	RO	0x-----00-	32	*Debug Base Address register, BASE* on page 9-217
0x0DFC	IDR	RO	0x04770014	32	*Identification Register, IDR* on page 9-218
0x0EFC	ITSTATUS	RW	0x00000000	32	*Integration Test Status register, ITSTATUS* on page 9-219
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-220

Table 9-87 css600_axiap - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FA0	CLAIMSET	RW	0x00000003	32	*Claim Tag Set Register, CLAIMSET* on page 9-221
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-222
0x0FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-223
0x0FBC	DEVARCH	RO	0x47700a17	32	*Device Architecture Register, DEVARCH* on page 9-225
0x0FCC	DEVTYPE	RO	0x00000000	32	*Device Type Identifier Register, DEVTYPE* on page 9-226
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-227
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-228
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-229
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-230
0x0FE0	PIDR0	RO	0x000000e4	32	*Peripheral Identification Register 0, PIDR0* on page 9-231
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-232
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-233
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-234
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-235
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-236
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-237
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-238
0x1000	DAR0	RW	0x--------	32	*Direct Access Register 0, DAR0* on page 9-200
0x1004	DAR1	RW	0x--------	32	*Direct Access Register 1, DAR1* on page 9-201
0x1008	DAR2	RW	0x--------	32	*Direct Access Register 2, DAR2* on page 9-202
…	…	…	…	…	…
0x13FC	DAR255	RW	0x--------	32	*Direct Access Register 255, DAR255* on page 9-203
0x1D00	CSW	RW	0x30-060-2	32	*Control Status Word register, CSW* on page 9-204
0x1D04	TAR	RW	0x--------	32	*Transfer Address Register, TAR* on page 9-206
0x1D08	TARH	RW	0x00000000	32	*Transfer Address Register, TARH* on page 9-207
0x1D0C	DRW	RW	0x--------	32	*Data Read/Write register, DRW* on page 9-208
0x1D10	BD0	RW	0x--------	32	*Banked Data register 0, BD0* on page 9-209
0x1D14	BD1	RW	0x--------	32	*Banked Data register 1, BD1* on page 9-210
0x1D18	BD2	RW	0x--------	32	*Banked Data register 2, BD2* on page 9-211
0x1D1C	BD3	RW	0x--------	32	*Banked Data register 3, BD3* on page 9-212
0x1D20	MBT	RW	0x00000000	32	*Memory Barrier Transfer register, MBT* on page 9-213
0x1D24	TRR	RW	0x00000000	32	*Transfer Response Register, TRR* on page 9-214
0x1DF0	BASEH	RW	0x--------	32	*Debug Base Address register upper 32 bits, BASEH* on page 9-215
0x1DF4	CFG	RO	0x000101a0	32	*Configuration register, CFG* on page 9-216
0x1DF8	BASE	RO	0x-----00-	32	*Debug Base Address register, BASE* on page 9-217

Table 9-87 css600_axiap - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x1DFC	IDR	RO	0x04770014	32	*Identification Register, IDR* on page 9-218
0x1EFC	ITSTATUS	RW	0x00000000	32	*Integration Test Status register, ITSTATUS* on page 9-219
0x1F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-220
0x1FA0	CLAIMSET	RW	0x00000003	32	*Claim Tag Set Register, CLAIMSET* on page 9-221
0x1FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-222
0x1FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-223
0x1FBC	DEVARCH	RO	0x47700a17	32	*Device Architecture Register, DEVARCH* on page 9-225
0x1FCC	DEVTYPE	RO	0x00000000	32	*Device Type Identifier Register, DEVTYPE* on page 9-226
0x1FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-227
0x1FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-228
0x1FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-229
0x1FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-230
0x1FE0	PIDR0	RO	0x000000e4	32	*Peripheral Identification Register 0, PIDR0* on page 9-231
0x1FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-232
0x1FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-233
0x1FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-234
0x1FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-235
0x1FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-236
0x1FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-237
0x1FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-238

9.5.2 Register descriptions

This section describes the `css600_axiap` registers.

9.5.1 Register summary on page 9-197 provides cross references to individual registers.

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.

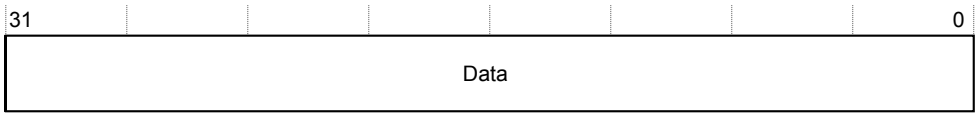

Figure 9-84 DAR0 register bit assignments

The following table shows the bit assignments.

Table 9-88 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.

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.

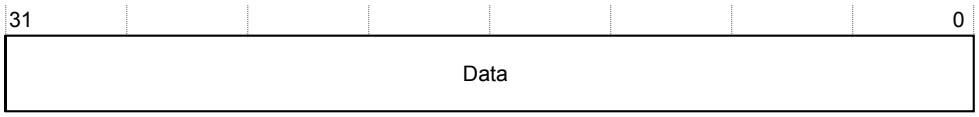

Figure 9-85 DAR1 register bit assignments

The following table shows the bit assignments.

Table 9-89 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.

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.

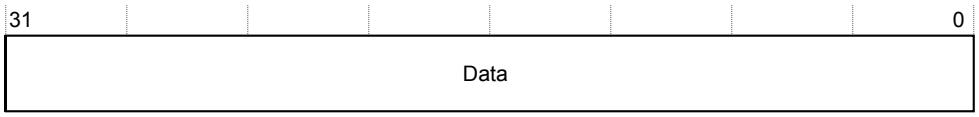

Figure 9-86 DAR2 register bit assignments

The following table shows the bit assignments.

Table 9-90 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.

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.

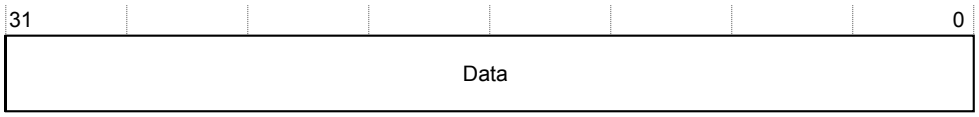

Figure 9-87 DAR255 register bit assignments

The following table shows the bit assignments.

Table 9-91 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.

Control Status Word register, CSW

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

The CSW register characteristics are:

Attributes

Offset	`0x0d00`
Type	Read-write
Reset	`0x30-060-2`
Width	32

The following figure shows the bit assignments.

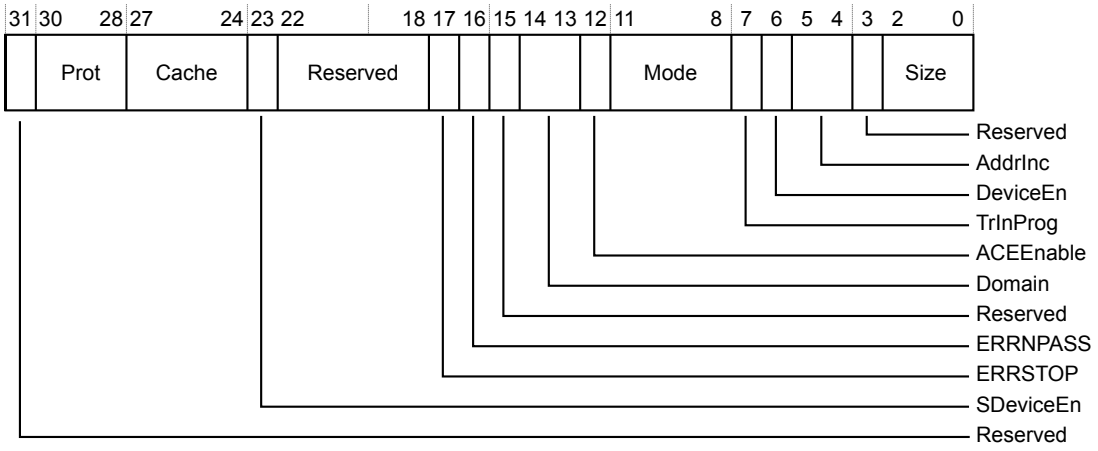

Figure 9-88 CSW register bit assignments

The following table shows the bit assignments.

Table 9-92 CSW register bit assignments

Bits	Reset value	Name	Function
[30:28]	`0b011`	Prot	Drives AXI master interface ports **arprot_m[2:0]** and **awprot_m[2:0]** which specifies the AXI4 protection encoding. The reset value is `0x3` (Data, Non-secure, Privileged). Together with authentication interface signals CSW[1] determines whether a secure access is allowed on the master interface as follows, access = **dbgen** && **spiden** || **dbgen** && CSW[1].
[27:24]	`0b0000`	Cache	Specifies the AXI3 and AXI4 cache encodings. Software must never program an invalid combination of values in CSW.Cache and CSW.Domain fields. The software must use different cache encoding values for reads and writes. If an illegal set of cache and domain values is programmed, the AXI AP does not issue the transaction on its master interface and generates a memory access error.
[23]	UNKNOWN	SDeviceEn	Indicates the status of the **spiden** and **spniden** ports. It is set when either **spiden** or **spniden** is HIGH, and remains clear otherwise. If this bit is clear, Secure AXI 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.
[17]	`0b0`	ERRSTOP	Stop on error. **0** Memory access errors do not prevent future memory accesses. **1** Memory access errors prevent future memory accesses.

Table 9-92 CSW register bit assignments (continued)

Bits	Reset value	Name	Function
[16]	0b0	ERRNPASS	Errors that are not passed upstream. **0** Memory access errors that are passed upstream. **1** Memory access errors that are not passed upstream.
[14:13]	0b11	Domain	Shareable transaction encoding for ACE. 0x0 Non-shareable. 0x1 Shareable, inner domain, includes additional masters. 0x2 Shareable, outer domain, also includes inner or additional masters. 0x3 Shareable, system domain, all masters included.
[12]	0b0	ACEEnable	Enable ACE transactions, including barriers. **0** Disable **1** Enable
[11:8]	0b0000	Mode	Specifies the mode of operation. 0x0 Normal download or upload mode. 0x1 Barrier transaction. 0x2-0xF Reserved.
[7]	0b0	TrInProg	Transfer in progress. This field indicates whether a transfer is in progress on the AXI master port.
[6]	UNKNOWN	DeviceEn	Indicates the status of **dbgen** and **niden** ports. The bit is set when either **dbgen** or **niden** is high, and is clear otherwise. If this bit is clear, no AXI 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.
[2:0]	0b010	Size	Size of the data access to perform. 0x0 8 bits. 0x1 16 bits. 0x2 32 bits. 0x3 64 bits, if LDE is supported, Reserved, if LDE is not supported. 0x4 Reserved. 0x5 Reserved. 0x6 Reserved. 0x7 Reserved.

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.

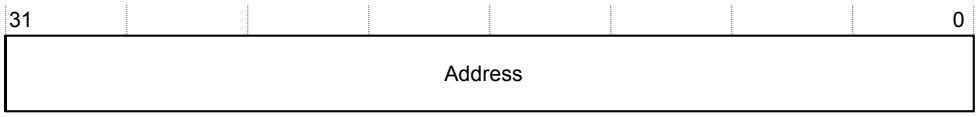

Figure 9-89 TAR register bit assignments

The following table shows the bit assignments.

Table 9-93 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.

Transfer Address Register, TARH

Holds the upper 32 bits of the Transfer Address Register if the AXIAP is configured to support LAE.

The TARH register characteristics are:

Attributes

Offset `0x0d08`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

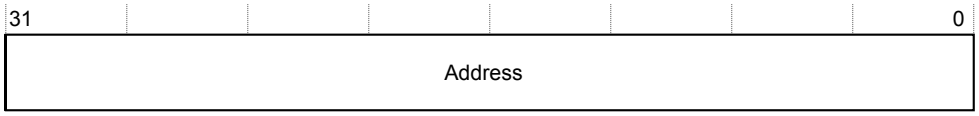

Figure 9-90 TARH register bit assignments

The following table shows the bit assignments.

Table 9-94 TARH register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	Address	Bits [63:32] of the transfer address. This register is present only when LAE is supported. Otherwise, it is reserved and RAZ/WI.

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.

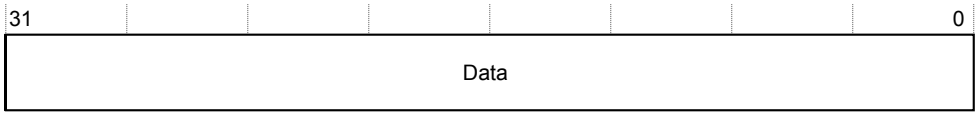

Figure 9-91 DRW register bit assignments

The following table shows the bit assignments.

Table 9-95 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.

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.

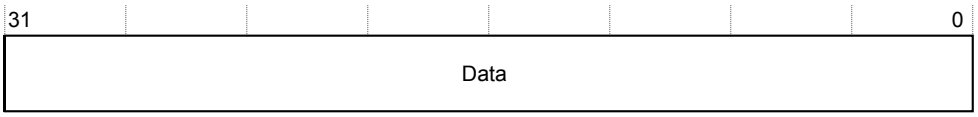

Figure 9-92 BD0 register bit assignments

The following table shows the bit assignments.

Table 9-96 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.

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.

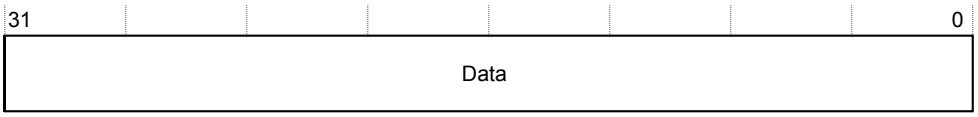

Figure 9-93 BD1 register bit assignments

The following table shows the bit assignments.

Table 9-97 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.

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.

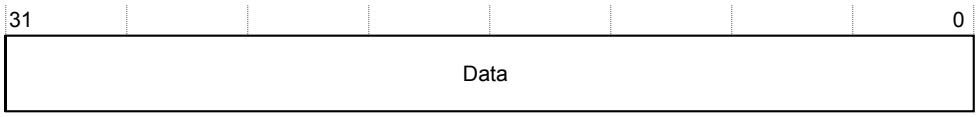

Figure 9-94 BD2 register bit assignments

The following table shows the bit assignments.

Table 9-98 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.

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.

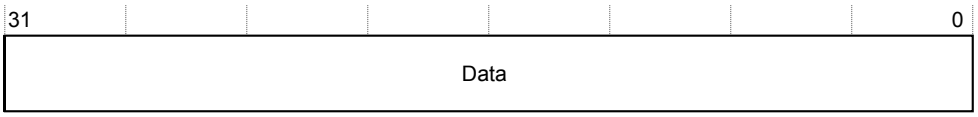

Figure 9-95 BD3 register bit assignments

The following table shows the bit assignments.

Table 9-99 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.

Memory Barrier Transfer register, MBT

Triggers a barrier transaction on the AXI interface. The write access on the APB slave interface is complete when the barrier transaction completes on the AXI interface. Until then, **pready_s** is held low. When the barrier transaction completes, TrgBarTran is cleared to 0. If another barrier transaction is required then you must write 1 to TrgBarTran again. If the TrgBarTran is already 1, then this write has no effect. It indicates that a barrier transaction is already in progress and has not completed. This is possible if an abort request aborted the earlier barrier transaction on the APB slave interface, and a new request to issue a barrier transaction is requested on the AXI interface. This results in an error response from the AXI-AP, if the barrier transaction is still not complete.

The MBT register characteristics are:

Attributes

Offset `0x0d20`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

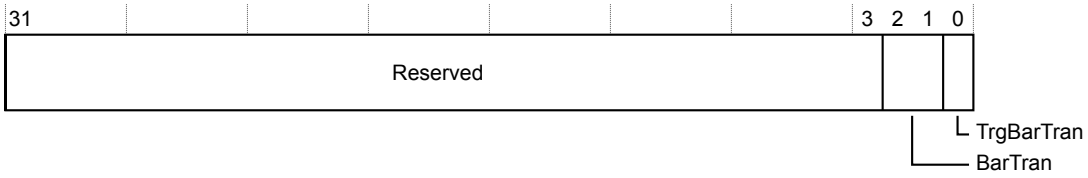

Figure 9-96 MBT register bit assignments

The following table shows the bit assignments.

Table 9-100 MBT register bit assignments

Bits	Reset value	Name	Function
[2:1]	`0b00`	BarTran	Barrier transactions. `0x0` Barrier with normal access. `0x1` Memory barrier. `0x2` Reserved. `0x3` Synchronization Barrier.
[0]	`0b0`	TrgBarTran	This bit triggers barrier transactions when written to 1.

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.

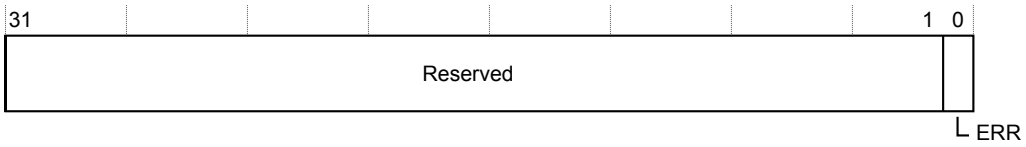

Figure 9-97 TRR register bit assignments

The following table shows the bit assignments.

Table 9-101 TRR register bit assignments

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.

Debug Base Address register upper 32 bits, BASEH

Holds the upper 32 bits of the base address of a ROM table when LAE is supported.

The BASEH register characteristics are:

Attributes

Offset `0x0df0`

Type Read-write

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

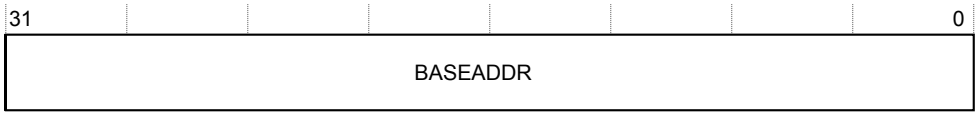

Figure 9-98 BASEH register bit assignments

The following table shows the bit assignments.

Table 9-102 BASEH register bit assignments

Bits	Reset value	Name	Function
[31:0]	IMPLEMENTATION_DEFINED	BASEADDR	Bits [63:32] of the ROM table base address. This register is present only when LAE is supported, otherwise it is reserved and RAZ/WI. Even with LAE supported, 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 port **baseaddr[63:32]**. Otherwise, it reads as `0x0`.

Configuration register, CFG

This is the AXIAP Configuration register.

The CFG register characteristics are:

Attributes

Offset	`0x0df4`
Type	Read-only
Reset	`0x000101a0`
Width	32

The following figure shows the bit assignments.

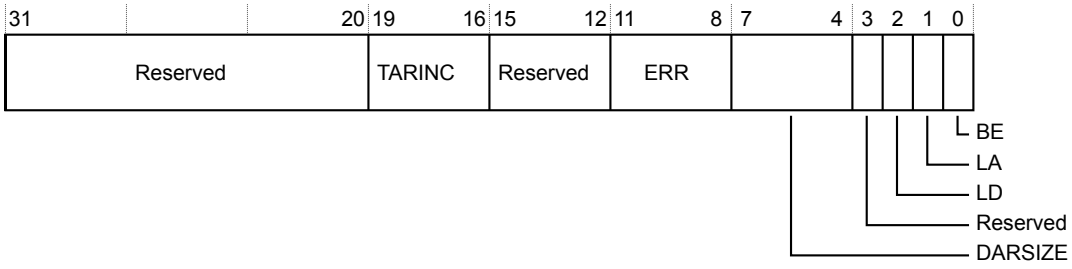

Figure 9-99 CFG register bit assignments

The following table shows the bit assignments.

Table 9-103 CFG register bit assignments

Bits	Reset value	Name	Function
[19:16]	`0b0001`	TARINC	TAR incrementer size. Returns `0x1` indicating a TAR incrementer size of 10-bits.
[11:8]	`0b0001`	ERR	Error functionality implemented. Returns `0x1` indicating that Error Response Handling version 1 is 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.
[2]	`0b0`	LD	Large Data. Indicates support for LDE (data items greater than 32 bits). This value of bit is fixed in a given configuration of AXI AP based on parameter `AXI_DATA_WIDTH`. **0** Only 8, 16, and 32-bit data items are supported. **1** Support for 64-bit data item in addition to 8, 16, and 32-bit data.
[1]	`0b0`	LA	Long Address. Indicates support for LAE (greater than 32-bit of addressing). This bit value is fixed in a given configuration of AXI AP based on parameter `AXI_ADDR_WIDTH`. **0** 32 or fewer bits of addressing. Registers `0xD08` and `0xDF0` are reserved. **1** 64 or fewer bits of addressing. TAR and BASE registers occupy two locations, at `0xD04` and `0xD08`, and at `0xDF8` and `0xDF0` respectively.
[0]	`0b0`	BE	Big-endian. Always read as 0 because AXI AP only supports little-endian.

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.

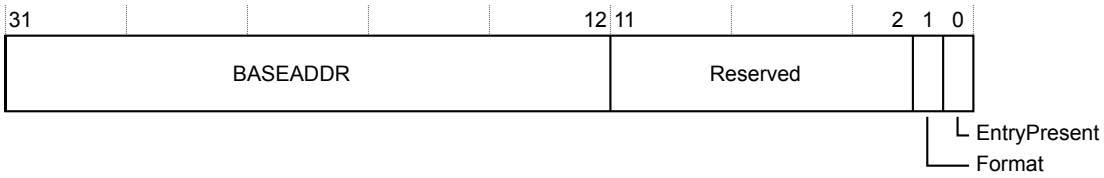

Figure 9-100 BASE register bit assignments

The following table shows the bit assignments.

Table 9-104 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 port **baseaddr[31:12]**, otherwise, it reads as `0x0`.
[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 port **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.

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	`0x04770014`
Width	32

The following figure shows the bit assignments.

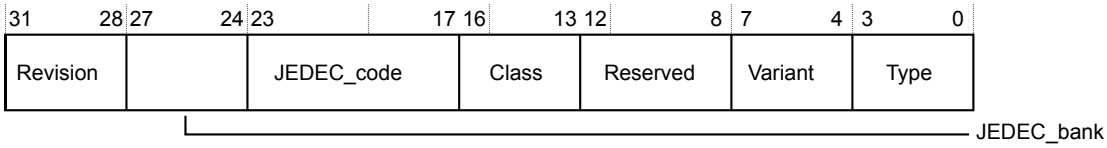

Figure 9-101 IDR register bit assignments

The following table shows the bit assignments.

Table 9-105 IDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	`0b0000`	Revision	Returns `0x0` (r0p0).
[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.
[7:4]	`0b0001`	Variant	Returns `0x1`, indicating variation from base type specified by IDR.Type.
[3:0]	`0b0100`	Type	Returns `0x4`, indicating that this is an AXI3 or AXI4 with optional ACE-Lite support Access Port.

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.

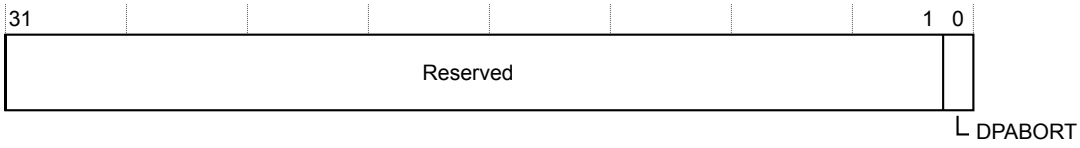

Figure 9-102 ITSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-106 ITSTATUS register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	DPABORT	When in Integration testing mode (f.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.

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.

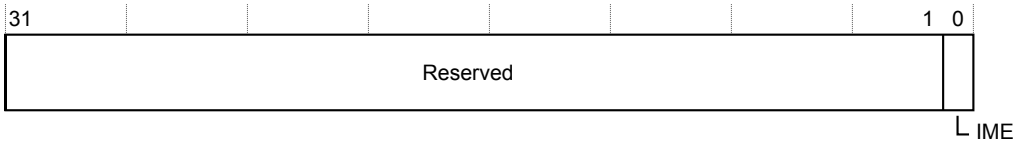

Figure 9-103 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-107 ITCTRL register bit assignments

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

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.

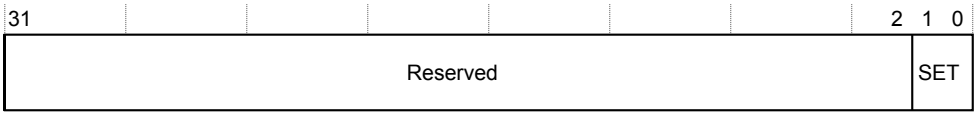

Figure 9-104 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-108 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[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.

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.

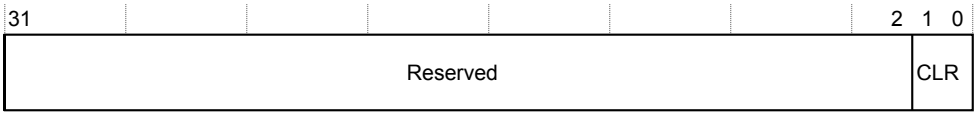

Figure 9-105 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-109 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[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.

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.

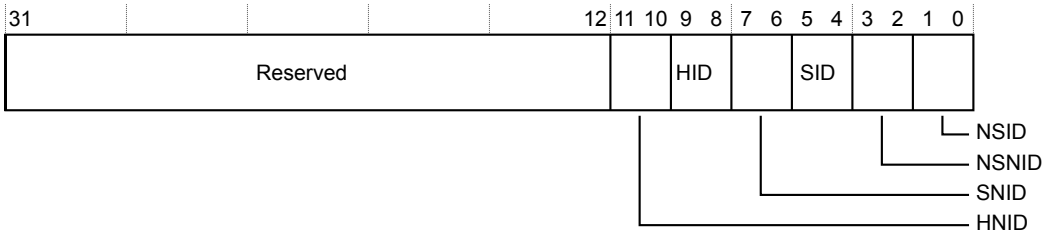

Figure 9-106 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-110 AUTHSTATUS register bit assignments

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.

Table 9-110 AUTHSTATUS register bit assignments (continued)

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.

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.

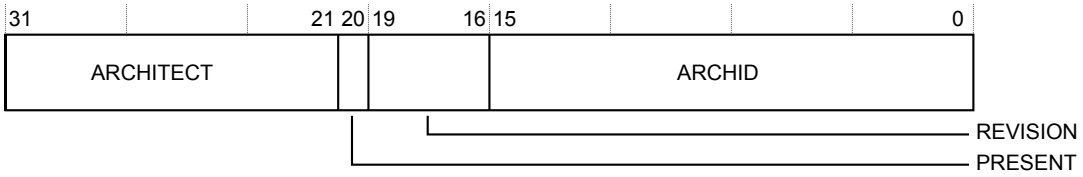

Figure 9-107 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-111 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.

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.

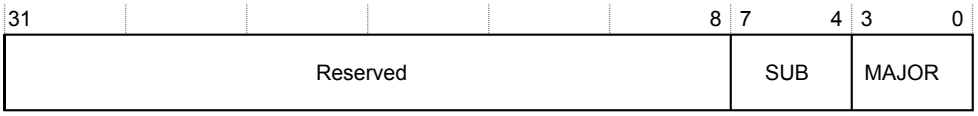

Figure 9-108 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-112 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	SUB	Minor classification. Returns `0x0`, Other/undefined.
[3:0]	`0b0000`	MAJOR	Major classification. Returns `0x0`, Miscellaneous.

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.

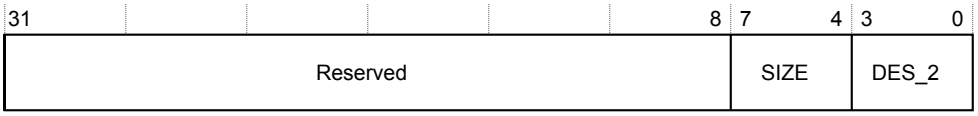

Figure 9-109 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-113 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

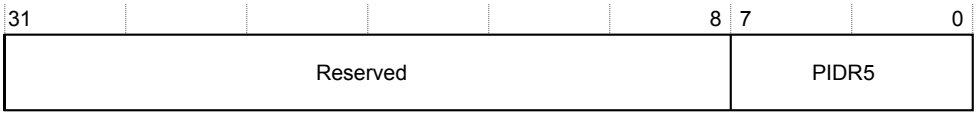

Figure 9-110 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-114 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

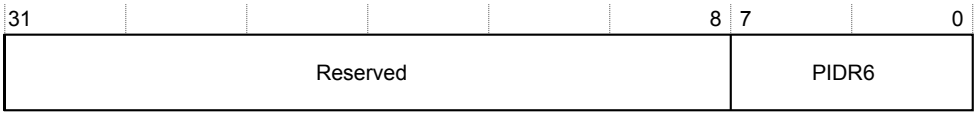

Figure 9-111 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-115 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

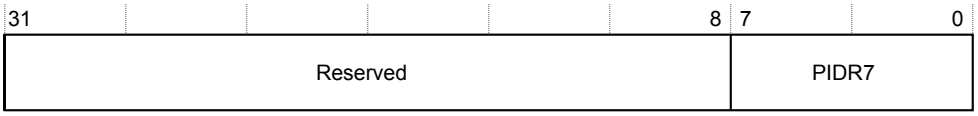

Figure 9-112 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-116 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000e4`

Width 32

The following figure shows the bit assignments.

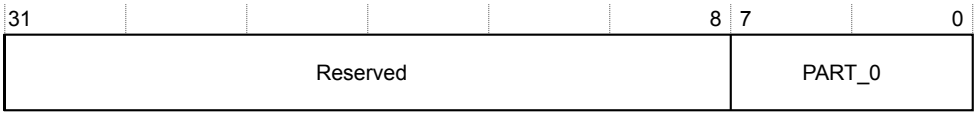

Figure 9-113 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-117 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11100100`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

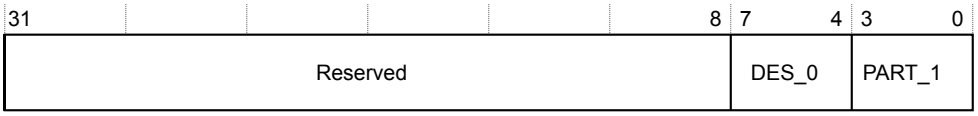

Figure 9-114 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-118 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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	`0x0000000b`
Width	32

The following figure shows the bit assignments.

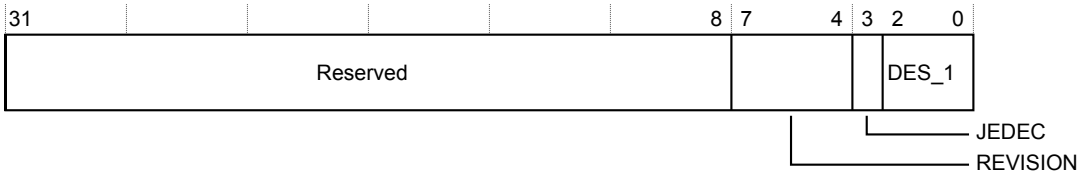

Figure 9-115 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-119 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

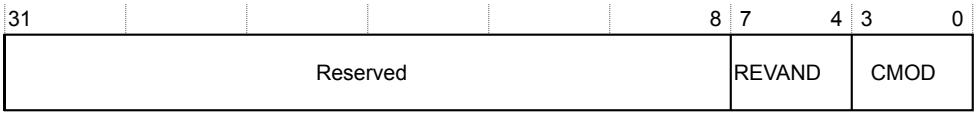

Figure 9-116 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-120 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

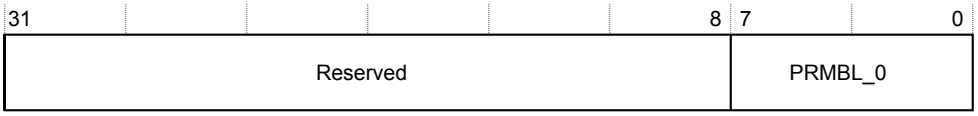

Figure 9-117 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-121 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

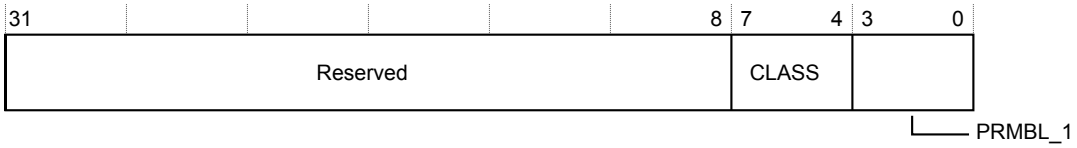

Figure 9-118 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-122 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

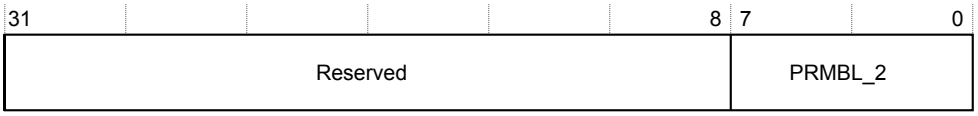

Figure 9-119 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-123 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

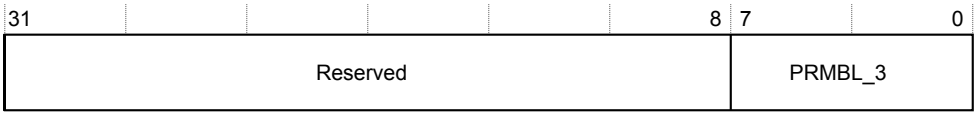

Figure 9-120 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-124 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.6 css600_apv1adapter introduction

This section describes the functions and programmers model of the `css600_apv1adapter`.

This section contains the following subsections:

- *9.6.1 Register summary* on page 9-239.
- *9.6.2 Register descriptions* on page 9-240.

9.6.1 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.

Table 9-125 css600_apv1adapter - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0D00	Downstream reg 0	IMPLEMENTATION_DEFINED	`0x--------`	32	Accesses AP1 register at 0x00000000.
0x0D04	Downstream reg 1	IMPLEMENTATION_DEFINED	`0x--------`	32	Accesses AP1 register at 0x00000004.
0x0D08	Downstream reg 2	IMPLEMENTATION_DEFINED	`0x--------`	32	Accesses AP1 register at 0x00000008.
…	…	…	…	…	…
0x0DFC	Downstream reg 63	IMPLEMENTATION_DEFINED	`0x--------`	32	Accesses AP1 register at 0x000000FC.
`0x0EFC`	ITSTATUS	RW	`0x00000000`	32	*Integration Test Status register, ITSTATUS* on page 9-241
`0x0F00`	ITCTRL	RW	`0x00000000`	32	*Integration Mode Control Register, ITCTRL* on page 9-242
`0x0FA0`	CLAIMSET	RW	`0x00000003`	32	*Claim Tag Set Register, CLAIMSET* on page 9-243
`0x0FA4`	CLAIMCLR	RW	`0x00000000`	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-244
`0x0FB8`	AUTHSTATUS	RO	`0x00000000`	32	*Authentication Status Register, AUTHSTATUS* on page 9-245
`0x0FBC`	DEVARCH	RO	`0x47700a47`	32	*Device Architecture Register, DEVARCH* on page 9-247
`0x0FCC`	DEVTYPE	RO	`0x00000000`	32	*Device Type Identifier Register, DEVTYPE* on page 9-248
`0x0FD0`	PIDR4	RO	`0x00000004`	32	*Peripheral Identification Register 4, PIDR4* on page 9-249
`0x0FD4`	PIDR5	RO	`0x00000000`	32	*Peripheral Identification Register 5, PIDR5* on page 9-250
`0x0FD8`	PIDR6	RO	`0x00000000`	32	*Peripheral Identification Register 6, PIDR6* on page 9-251

Table 9-125 css600_apv1adapter - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-252
0x0FE0	PIDR0	RO	0x000000e5	32	*Peripheral Identification Register 0, PIDR0* on page 9-253
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-254
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-255
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-256
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-257
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-258
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-259
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-260

9.6.2 Register descriptions

This section describes the `css600_apv1adapter` registers.

9.6.1 Register summary on page 9-239 provides cross references to individual registers.

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.

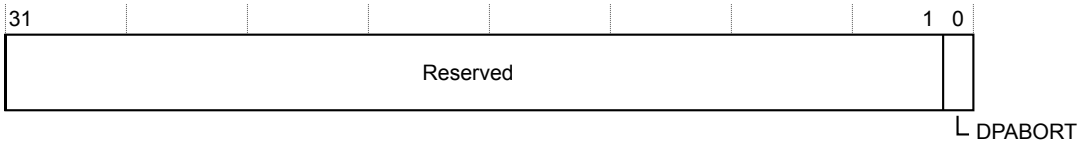

Figure 9-121 ITSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-126 ITSTATUS register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	DPABORT	When in Integration testing mode (f.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.

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.

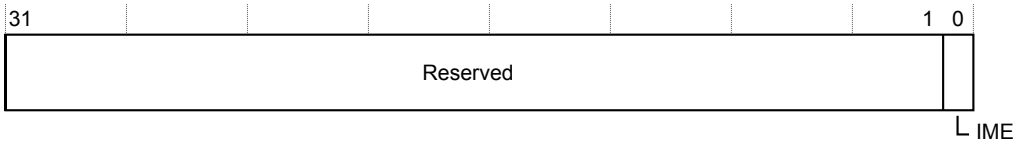

Figure 9-122 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-127 ITCTRL register bit assignments

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

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.

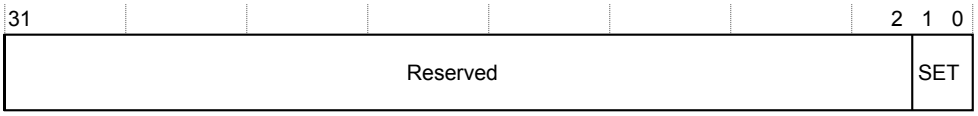

Figure 9-123 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-128 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[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.

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.

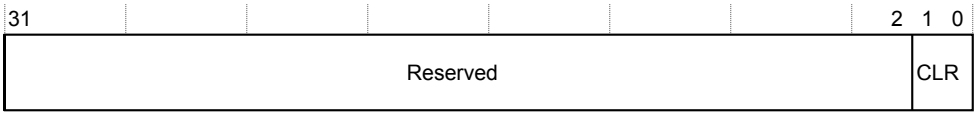

Figure 9-124 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-129 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[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.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

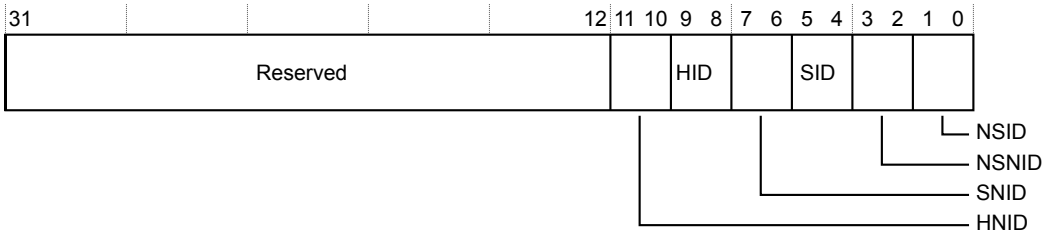

Figure 9-125 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-130 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	`0b00`	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-130 AUTHSTATUS register bit assignments (continued)

Bits	Reset value	Name	Function
[3:2]	0b00	NSNID	Non-secure non-invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.
[1:0]	0b00	NSID	Non-secure invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.

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 `0x47700a47`
Width 32

The following figure shows the bit assignments.

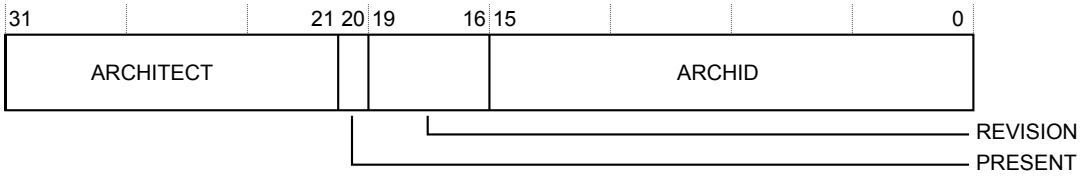

Figure 9-126 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-131 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]	`0xa47`	ARCHID	Architecture ID. Returns `0x0a47`, identifying Unknown AP.

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.

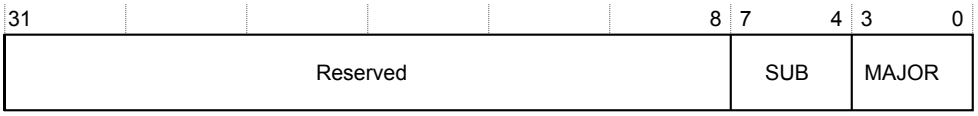

Figure 9-127 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-132 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	SUB	Minor classification. Returns `0x0`, Other/undefined.
[3:0]	`0b0000`	MAJOR	Major classification. Returns `0x0`, Miscellaneous.

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.

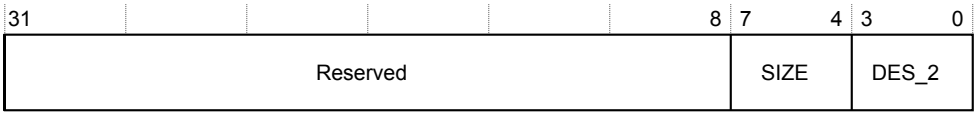

Figure 9-128 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-133 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

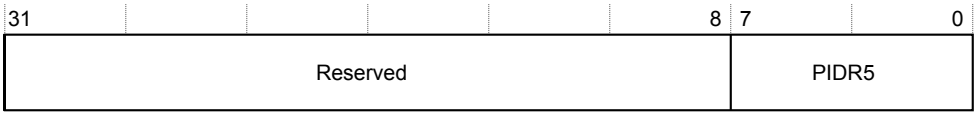

Figure 9-129 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-134 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

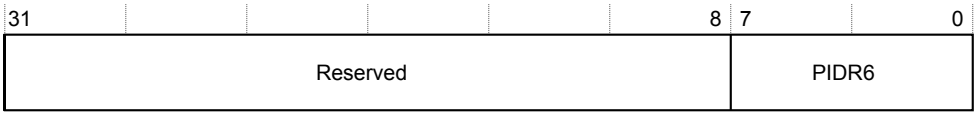

Figure 9-130 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-135 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

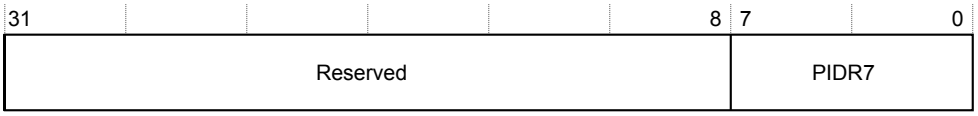

Figure 9-131 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-136 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	0b00000000	PIDR7	Reserved.

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 `0x000000e5`

Width 32

The following figure shows the bit assignments.

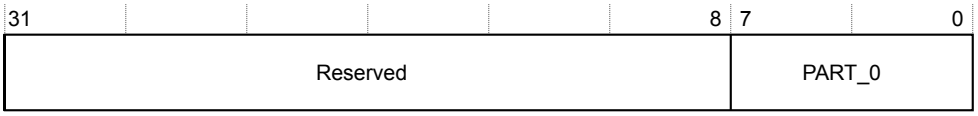

Figure 9-132 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-137 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11100101`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

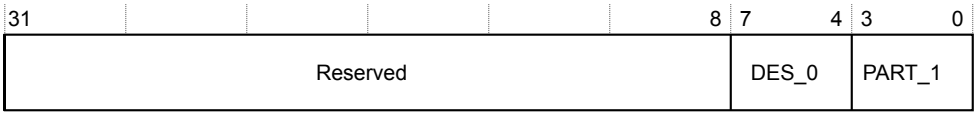

Figure 9-133 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-138 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x0000000b`
Width 32

The following figure shows the bit assignments.

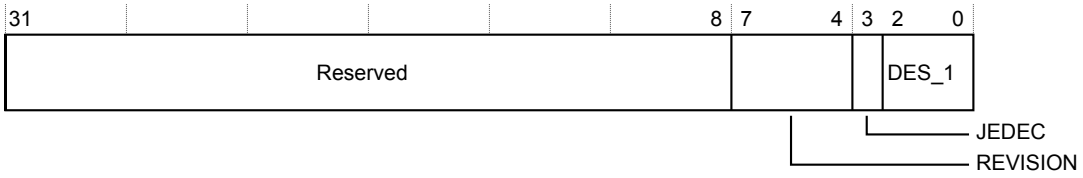

Figure 9-134 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-139 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

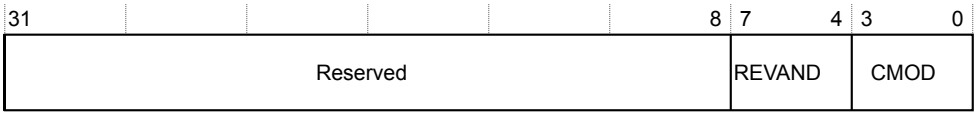

Figure 9-135 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-140 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

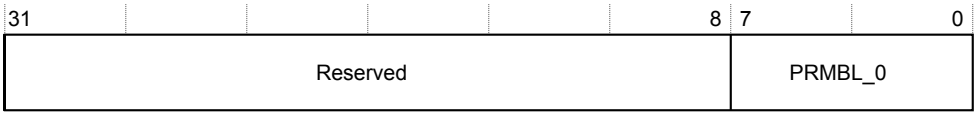

Figure 9-136 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-141 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

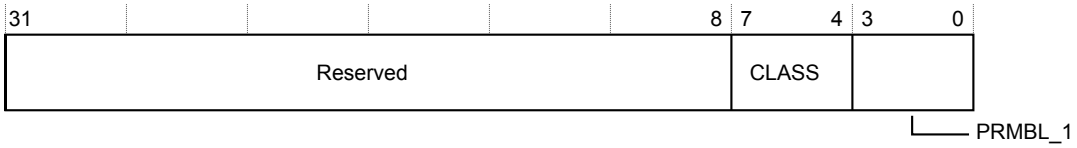

Figure 9-137 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-142 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

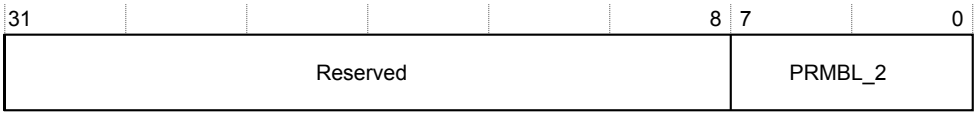

Figure 9-138 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-143 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

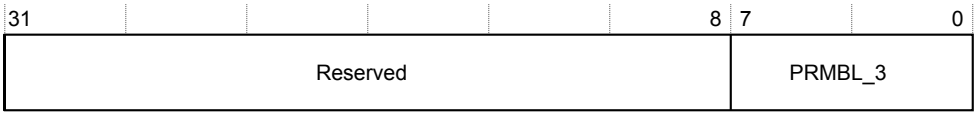

Figure 9-139 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-144 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.7 css600_jtagap introduction

This section describes the functions and programmers model of the `css600_jtagap`.

This section contains the following subsections:

- *9.7.1 Register summary* on page 9-261.
- *9.7.2 Register descriptions* on page 9-262.

9.7.1 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.

Table 9-145 css600_jtagap - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0D00	CSW	RW	0x00000000	32	*Control/Status Word register, CSW* on page 9-263
0x0D04	PSEL	RW	0x00000000	32	*Port Select register, PSEL* on page 9-265
0x0D08	PSTA	RW	0x00000000	32	*Port Status register, PSTA* on page 9-266
0x0D10	BFIFO1	RW	0x--------	32	*Byte FIFO Registers, BFIFO1* on page 9-267
0x0D14	BFIFO2	RW	0x--------	32	*Byte FIFO Registers, BFIFO2* on page 9-268
0x0D18	BFIFO3	RW	0x--------	32	*Byte FIFO Registers, BFIFO3* on page 9-269
0x0D1C	BFIFO4	RW	0x--------	32	*Byte FIFO Registers, BFIFO4* on page 9-270
0x0DFC	IDR	RO	0x04760020	32	*Identification Register, IDR* on page 9-271
0x0EFC	ITSTATUS	RW	0x00000000	32	*Integration Test Status register, ITSTATUS* on page 9-272
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-273
0x0FA0	CLAIMSET	RW	0x00000003	32	*Claim Tag Set Register, CLAIMSET* on page 9-274
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-275
0x0FBC	DEVARCH	RO	0x47700a27	32	*Device Architecture Register, DEVARCH* on page 9-276
0x0FCC	DEVTYPE	RO	0x00000000	32	*Device Type Identifier Register, DEVTYPE* on page 9-277
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-278
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-279
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-280
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-281
0x0FE0	PIDR0	RO	0x000000e6	32	*Peripheral Identification Register 0, PIDR0* on page 9-282
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-283
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-284
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-285
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-286

Table 9-145 css600_jtagap - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-287
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-288
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-289

9.7.2 Register descriptions

This section describes the `css600_jtagap` registers.

9.7.1 Register summary on page 9-261 provides cross references to individual registers.

Control/Status Word register, CSW

The CSW register configures and controls transfers through the JTAG interface to the connected memory system.

The CSW register characteristics are:

Attributes

Offset `0x0d00`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

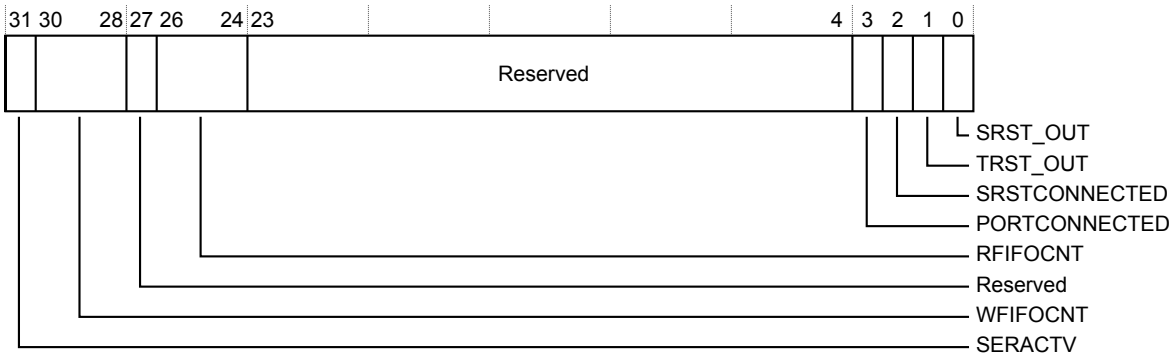

Figure 9-140 CSW register bit assignments

The following table shows the bit assignments.

Table 9-146 CSW register bit assignments

Bits	Reset value	Name	Function
[31]	`0b0`	SERACTV	JTAG engine active. This bit gets set when the JTAG engine picks the first command from the Command FIFO for execution and remains set until all commands have been executed, that is until after CSW.WFIFOCNT becomes 0 and the JTAG engine goes to idle state. **0** JTAG engine is inactive. **1** JTAG engine is processing commands from the Command FIFO.
[30:28]	`0b000`	WFIFOCNT	Command FIFO outstanding byte count. The reset value is `0x0`. Returns the number of command bytes held in the Command FIFO that are yet to be processed by the JTAG engine. Since the Command FIFO is 4 entries deep, this field can only take values between 0 and 4.
[26:24]	`0b000`	RFIFOCNT	Response FIFO outstanding byte count. The reset value is `0x0`. Returns the number of bytes of response data held in the Response FIFO. Since the Response FIFO is 7 entries deep, this field can take any value between 0 and 7.
[3]	`0b0`	PORTCONNECTED	PORT connected. This bit indicates the logical AND of **port_connected** inputs from all ports that are currently selected in the PSEL register.
[2]	`0b0`	SRSTCONNECTED	SRST connected. This bit is logical AND of **srst_connected** inputs from all ports that are currently selected in PSEL register.

Table 9-146 CSW register bit assignments (continued)

Bits	Reset value	Name	Function
[1]	0b0	TRST_OUT	This bit specifies the value to drive out on the active-LOW **cs_ntrst** pin for the ports that are connected, selected, and their PSTA bit is clear. This bit does not self-clear and must be cleared by a software write to this register. **0** De-assert **cs_ntrst** HIGH. **1** Assert **cs_ntrst** LOW.
[0]	0b0	SRST_OUT	This bit specifies the value to drive out on the active-LOW **srst_out_n** pin for the ports that are connected, selected, and their PSTA bit is clear. This bit does not self-clear and must be cleared by a software write to this register. **0** De-assert **srst_out_n** HIGH. **1** Assert **srst_out_n** LOW.

Port Select register, PSEL

Port Select register enables JTAG ports, provided the slave port is connected to the JTAG AP and **port_enabled** signal from the slave port to the JTAG AP is asserted HIGH. The port select register must be written only when the following conditions are met: the JTAG engine is idle AND the write FIFO is empty. If this register is written to in any other state, the corresponding JTAG ports are abruptly enabled, or disabled, in the middle of a transfer, which might cause errors, stalls, or deadlocks in the JTAG slave.

The PSEL register characteristics are:

Attributes

Offset `0x0d04`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

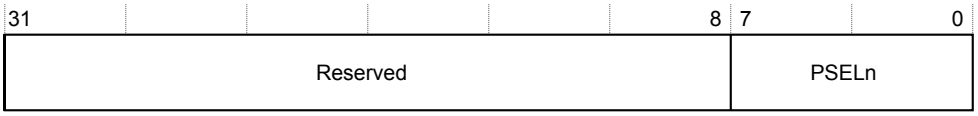

Figure 9-141 PSEL register bit assignments

The following table shows the bit assignments.

Table 9-147 PSEL register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PSELn	Port Select. Each register field is named as PSELn, where n = 0-7. The numerical index represents the bit position of that field in this register.

Port Status register, PSTA

The Port Status register captures the state of a connected and selected port on every clock cycle. If a connected and selected port is disabled or powered down, that is signal **port_enabled** goes low, even transiently, the corresponding bit in the PSTA register is set in the next cycle. It remains 1 until it is cleared by writing 1 to it. It gets cleared automatically on abort. Deselecting a port in PSEL does not alter the state of PSTA. If the PSTA bit is set for a port, that port is disabled and its TCK, TMS, and TDI outputs are driven LOW until its PSTA bit is cleared. Software must not clear any PSTA bit unless the JTAG-AP is idle, that is CSW.SERACTV=`0b0` and CSW.WFIFOCNT=`0x0`.

The PSTA register characteristics are:

Attributes

Offset `0x0d08`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

Figure 9-142 PSTA register bit assignments

The following table shows the bit assignments.

Table 9-148 PSTA register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PSTAn	Port Status. Each register field is named as PSTAn, where n = 0-7. The numerical index represents the bit position of that field in this register.

Byte FIFO Registers, BFIFO1

The Byte FIFO Registers, together, enable up to four bytes to be transacted with the Response or Command FIFO, by reading or writing the appropriate register.

The BFIFO1 register characteristics are:

Attributes

Offset `0x0d10`

Type Read-write

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

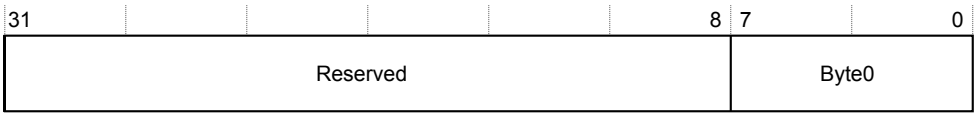

Figure 9-143 BFIFO1 register bit assignments

The following table shows the bit assignments.

Table 9-149 BFIFO1 register bit assignments

Bits	Reset value	Name	Function
[7:0]	UNKNOWN	Byte0	First byte.

Byte FIFO Registers, BFIFO2

The Byte FIFO Registers, together, enable up to four bytes to be transacted with the Response or Command FIFO, by reading or writing the appropriate register.

The BFIFO2 register characteristics are:

Attributes

Offset `0x0d14`

Type Read-write

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

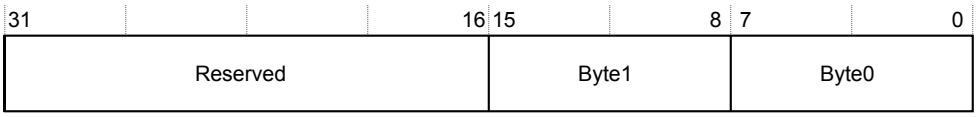

Figure 9-144 BFIFO2 register bit assignments

The following table shows the bit assignments.

Table 9-150 BFIFO2 register bit assignments

Bits	Reset value	Name	Function
[15:8]	UNKNOWN	Byte1	Second byte.
[7:0]	UNKNOWN	Byte0	First byte.

Byte FIFO Registers, BFIFO3

The Byte FIFO Registers, together, enable up to four bytes to be transacted with the Response or Command FIFO, by reading or writing the appropriate register.

The BFIFO3 register characteristics are:

Attributes

Offset `0x0d18`
Type Read-write
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

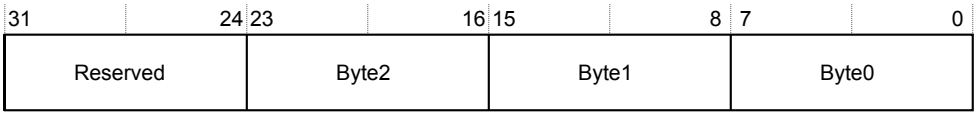

Figure 9-145 BFIFO3 register bit assignments

The following table shows the bit assignments.

Table 9-151 BFIFO3 register bit assignments

Bits	Reset value	Name	Function
[23:16]	UNKNOWN	Byte2	Third byte.
[15:8]	UNKNOWN	Byte1	Second byte.
[7:0]	UNKNOWN	Byte0	First byte.

Byte FIFO Registers, BFIFO4

The Byte FIFO Registers, together, enable up to four bytes to be transacted with the Response or Command FIFO, by reading or writing the appropriate register.

The BFIFO4 register characteristics are:

Attributes

Offset `0x0d1c`
Type Read-write
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

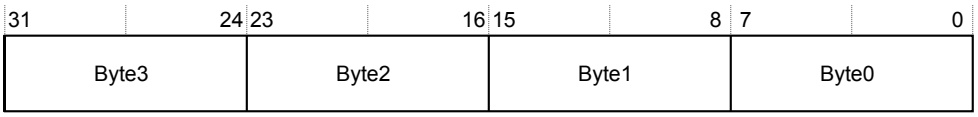

Figure 9-146 BFIFO4 register bit assignments

The following table shows the bit assignments.

Table 9-152 BFIFO4 register bit assignments

Bits	Reset value	Name	Function
[31:24]	UNKNOWN	Byte3	Forth byte.
[23:16]	UNKNOWN	Byte2	Third byte.
[15:8]	UNKNOWN	Byte1	Second byte.
[7:0]	UNKNOWN	Byte0	First byte.

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 0x04760020
Width 32

The following figure shows the bit assignments.

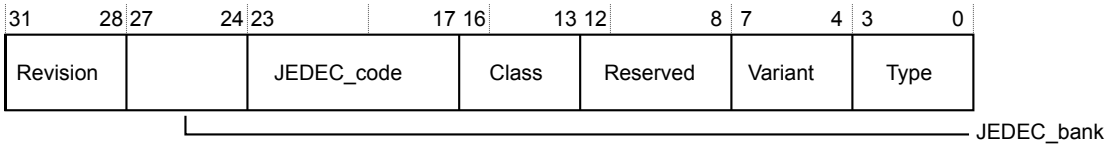

Figure 9-147 IDR register bit assignments

The following table shows the bit assignments.

Table 9-153 IDR register bit assignments

Bits	Reset value	Name	Function
[31:28]	0b0000	Revision	Returns 0x0 (r0p0).
[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.

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.

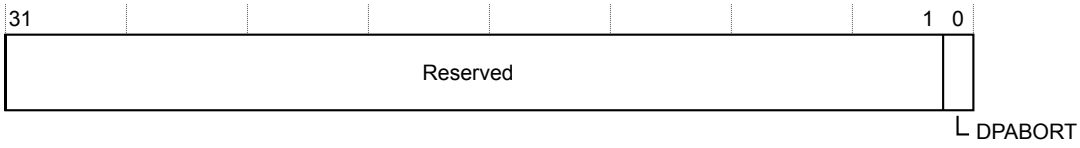

Figure 9-148 ITSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-154 ITSTATUS register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	DPABORT	When in Integration testing mode (f.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.

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.

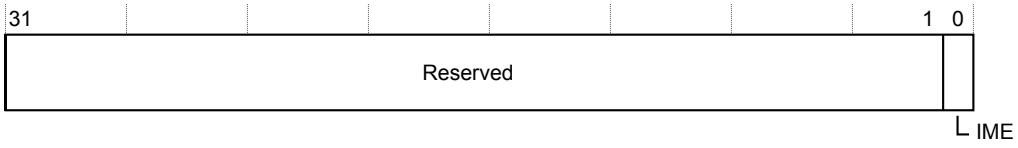

Figure 9-149 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-155 ITCTRL register bit assignments

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

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.

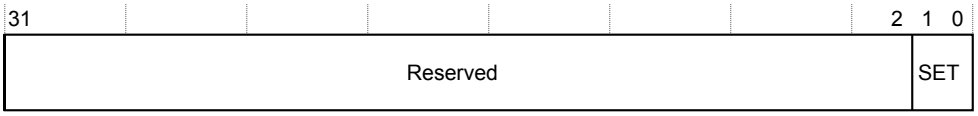

Figure 9-150 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-156 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[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.

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.

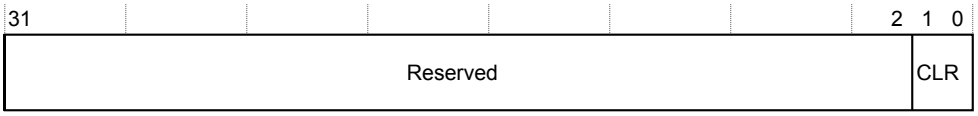

Figure 9-151 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-157 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[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.

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 `0x47700a27`
Width 32

The following figure shows the bit assignments.

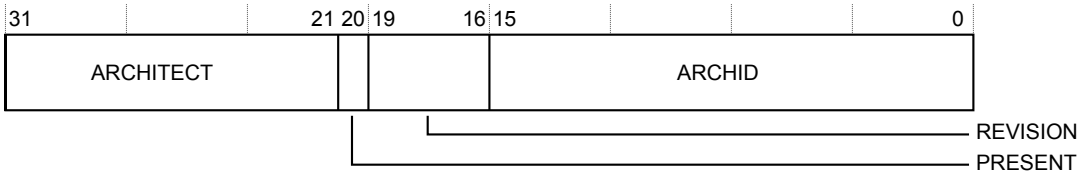

Figure 9-152 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-158 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]	`0xa27`	ARCHID	Architecture ID. Returns `0x0a27`, identifying APv2 JTAG-AP architecture v0.

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.

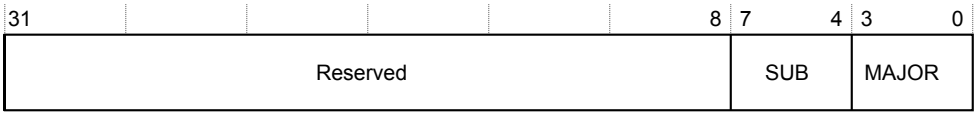

Figure 9-153 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-159 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	SUB	Minor classification. Returns `0x0`, Other/undefined.
[3:0]	`0b0000`	MAJOR	Major classification. Returns `0x0`, Miscellaneous.

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.

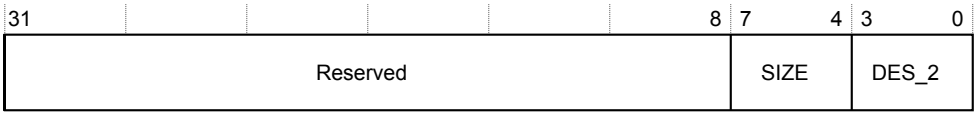

Figure 9-154 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-160 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

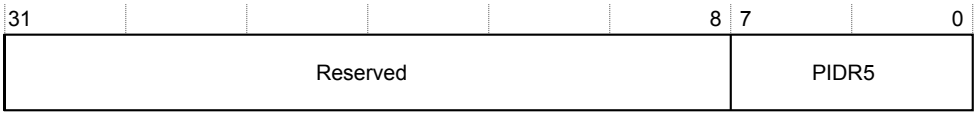

Figure 9-155 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-161 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

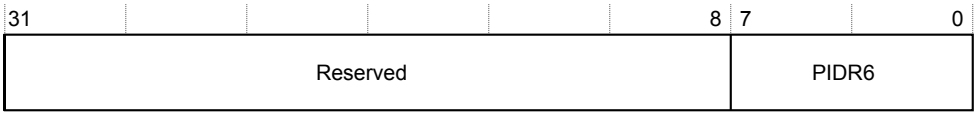

Figure 9-156 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-162 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

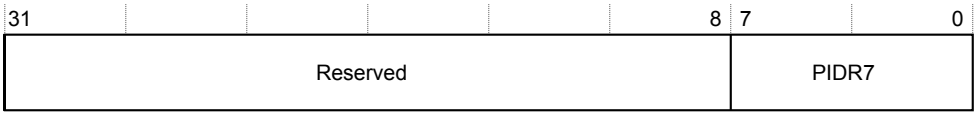

Figure 9-157 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-163 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	0b00000000	PIDR7	Reserved.

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 `0x000000e6`

Width 32

The following figure shows the bit assignments.

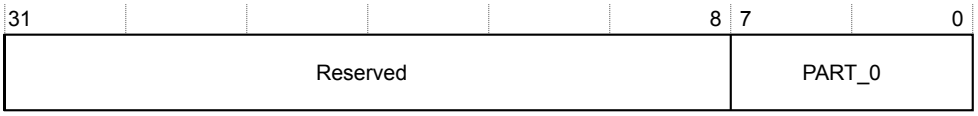

Figure 9-158 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-164 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11100110`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

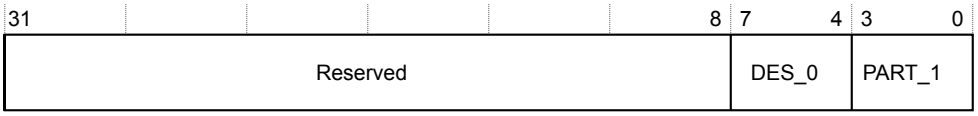

Figure 9-159 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-165 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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	`0x0000000b`
Width	32

The following figure shows the bit assignments.

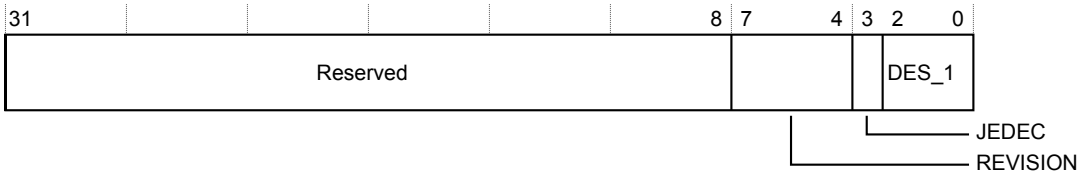

Figure 9-160 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-166 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

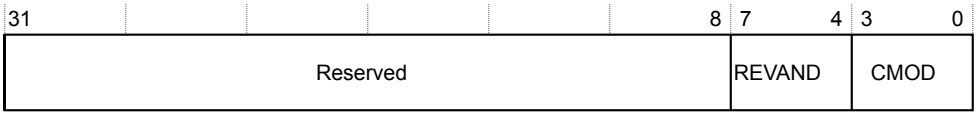

Figure 9-161 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-167 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

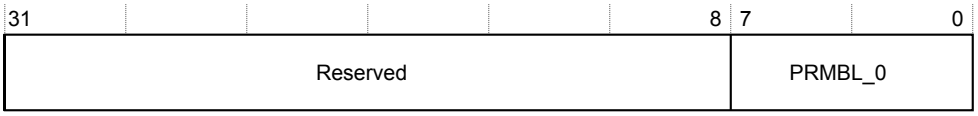

Figure 9-162 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-168 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

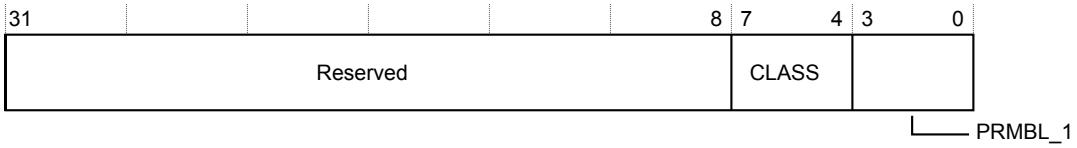

Figure 9-163 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-169 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

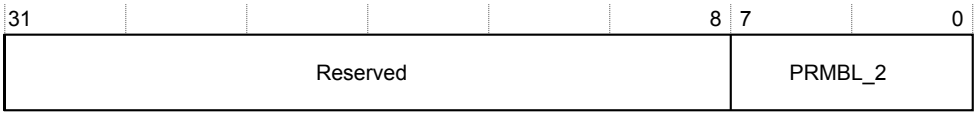

Figure 9-164 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-170 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

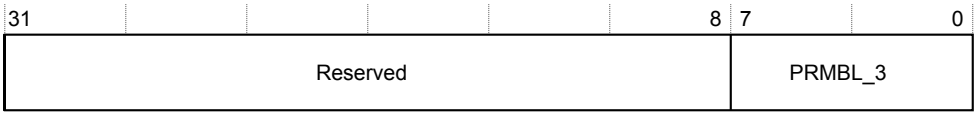

Figure 9-165 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-171 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.8 css600_apbrom introduction

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

This section contains the following subsections:

- *9.8.1 Register summary* on page 9-290.
- *9.8.2 Register descriptions* on page 9-290.

9.8.1 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.

Table 9-172 css600_apbrom - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	ROMEntry0	RO	0x--------	32	*ROM Entries register 0, ROMEntry0* on page 9-292
0x0004	ROMEntry1	RO	0x--------	32	*ROM Entries register 1, ROMEntry1* on page 9-293
0x0008	ROMEntry2	RO	0x--------	32	*ROM Entries register 2, ROMEntry2* on page 9-294
…	…	…	…	…	…
0x07FC	ROMEntry511	RO	0x--------	32	*ROM Entries register 511, ROMEntry511* on page 9-295
0x0FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-296
0x0FBC	DEVARCH	RO	0x47700af7	32	*Device Architecture Register, DEVARCH* on page 9-298
0x0FC8	DEVID	RO	0x000000-0	32	*Device Configuration Register, DEVID* on page 9-299
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-300
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-301
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-302
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-303
0x0FE0	PIDR0	RO	0x000000--	32	*Peripheral Identification Register 0, PIDR0* on page 9-304
0x0FE4	PIDR1	RO	0x000000--	32	*Peripheral Identification Register 1, PIDR1* on page 9-305
0x0FE8	PIDR2	RO	0x000000--	32	*Peripheral Identification Register 2, PIDR2* on page 9-306
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-307
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-308
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-309
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-310
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-311

9.8.2 Register descriptions

This section describes the `css600_apbrom` registers.

9.8.1 Register summary on page 9-290 provides cross references to individual registers.

Note

The ROM table has a configuration parameter `TIE_OFF_PRESENT`, which, if set to 1, allows for the selective removal of any entry using the **entry_present** input bus. The **entry_present** bus contains one bit per ROM table entry.

If a given **entry_present[n]** bit is tied HIGH, then the value in ROMEntry<n>.PRESENT[1:0] is taken directly from the value in the `ROM_ENTRY<n>` parameter for that entry.

If the **entry_present[n]** input is tied LOW, then a value of `0x3` in ROMEntry<n>.PRESENT[1:0] is modified to read `0x2` to indicate that the value is not present and is not the last entry.

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.

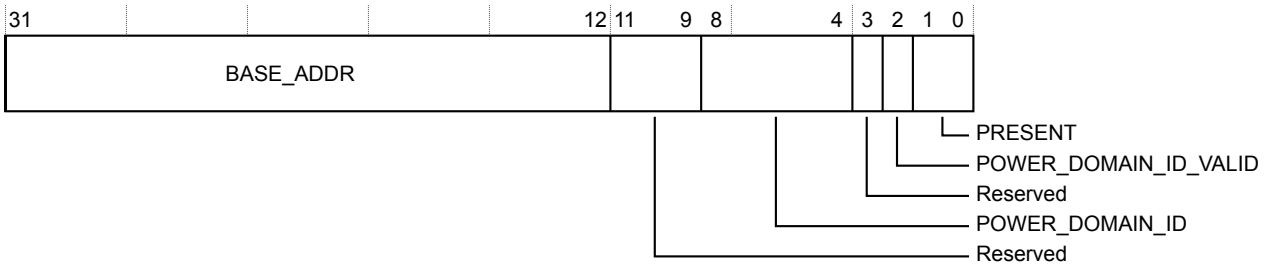

Figure 9-166 ROMEntry0 register bit assignments

The following table shows the bit assignments.

Table 9-173 ROMEntry0 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

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.

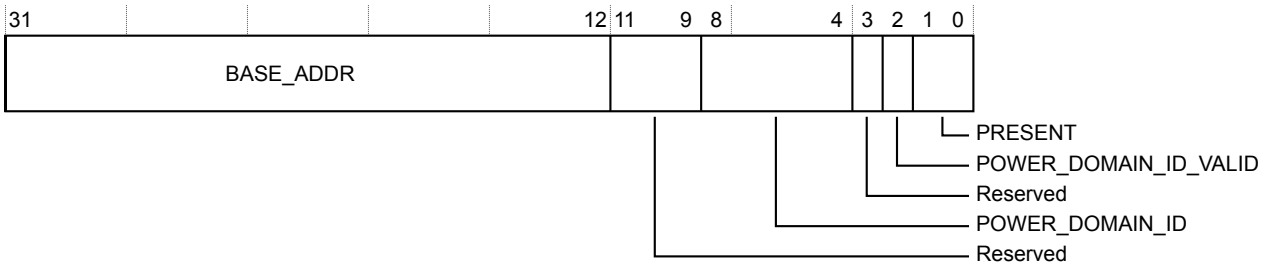

Figure 9-167 ROMEntry1 register bit assignments

The following table shows the bit assignments.

Table 9-174 ROMEntry1 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

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.

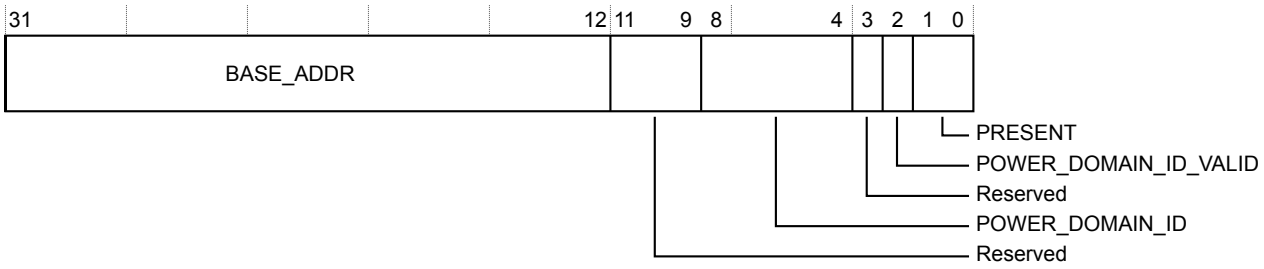

Figure 9-168 ROMEntry2 register bit assignments

The following table shows the bit assignments.

Table 9-175 ROMEntry2 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

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.

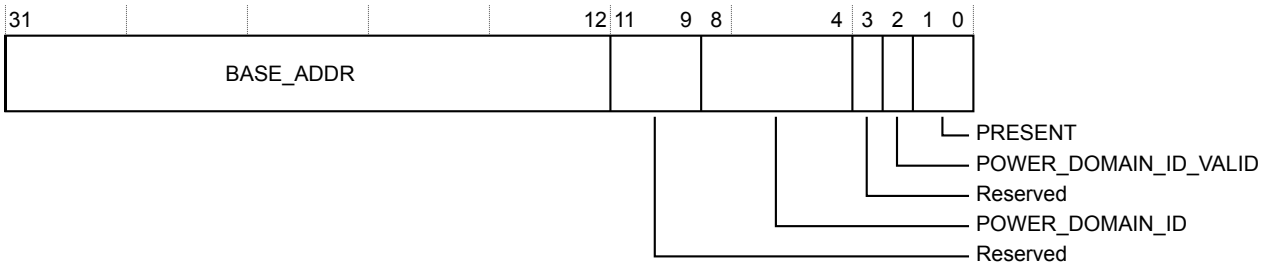

Figure 9-169 ROMEntry511 register bit assignments

The following table shows the bit assignments.

Table 9-176 ROMEntry511 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

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.

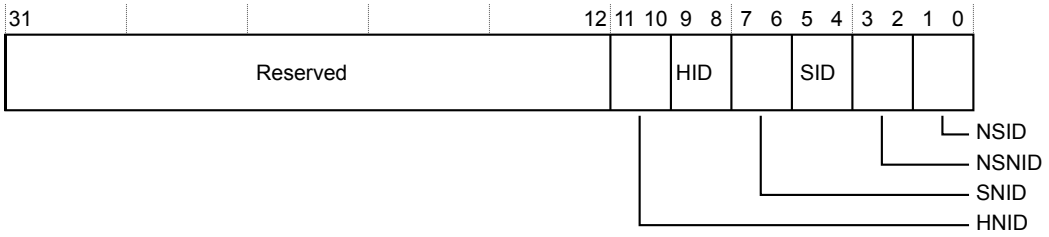

Figure 9-170 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-177 AUTHSTATUS register bit assignments

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.

Table 9-177 AUTHSTATUS register bit assignments (continued)

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.

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.

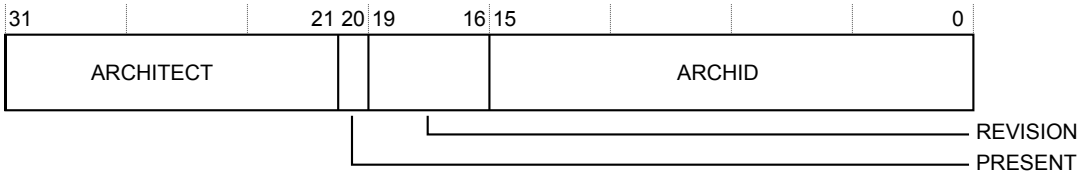

Figure 9-171 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-178 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.

Device Configuration Register, DEVID

This register is IMPLEMENTATION DEFINED for each Part Number and Designer. This 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.

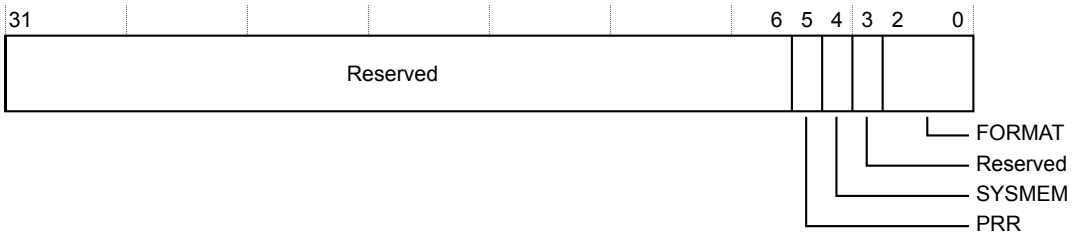

Figure 9-172 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-179 DEVID register bit assignments

Bits	Reset value	Name	Function
[5]	`0b1`	PRR	Indicates that power request functionality is included. Set by the GPR_PRESENT parameter. **0** GPR is not included (css600_rom). **1** GPR is included (css600_rom_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.
[2:0]	`0b000`	FORMAT	Indicates that this is a 32-bit ROM table.

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.

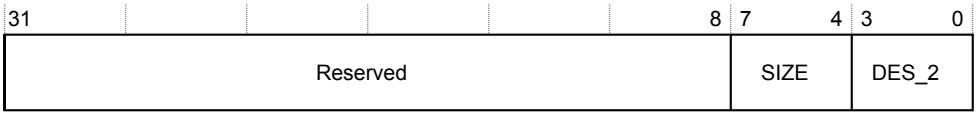

Figure 9-173 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-180 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

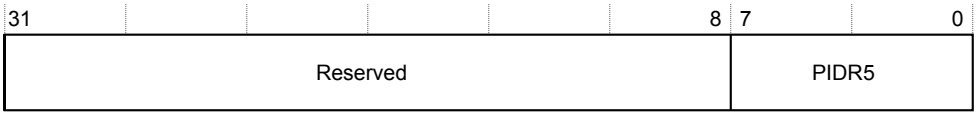

Figure 9-174 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-181 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

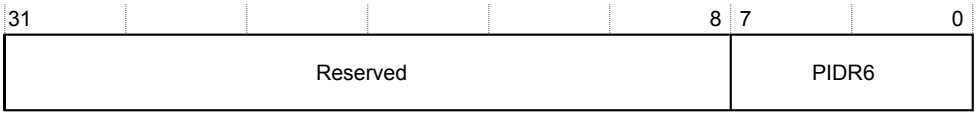

Figure 9-175 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-182 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

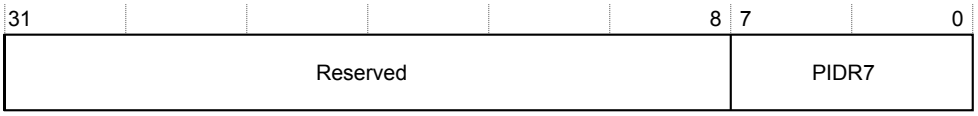

Figure 9-176 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-183 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	0b00000000	PIDR7	Reserved.

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.

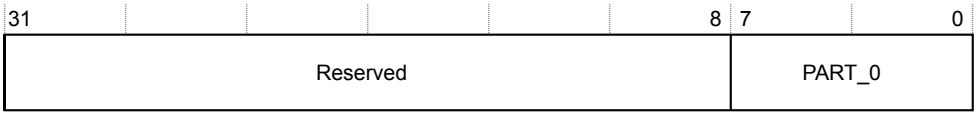

Figure 9-177 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-184 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	IMPLEMENTATION_DEFINED	PART_0	Part number, bits[7:0]. Set by the configuration inputs **part_number[7:0]**

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.

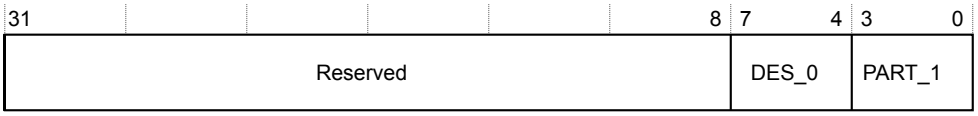

Figure 9-178 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-185 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]**

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.

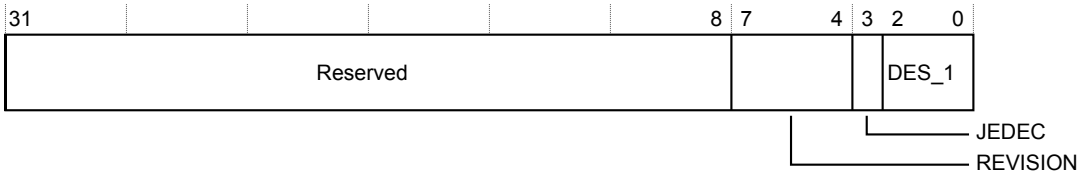

Figure 9-179 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-186 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

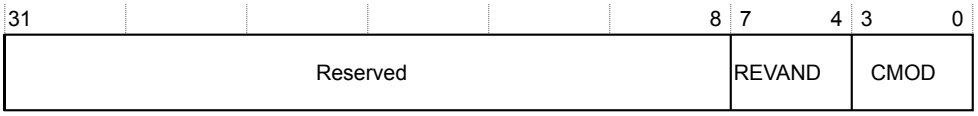

Figure 9-180 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-187 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

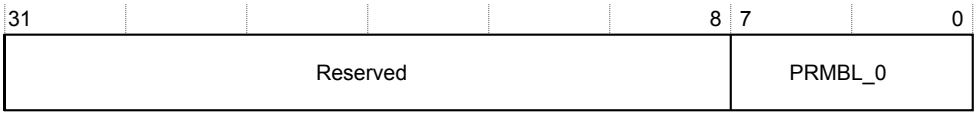

Figure 9-181 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-188 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

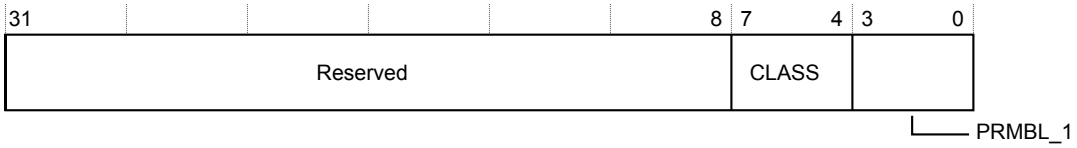

Figure 9-182 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-189 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

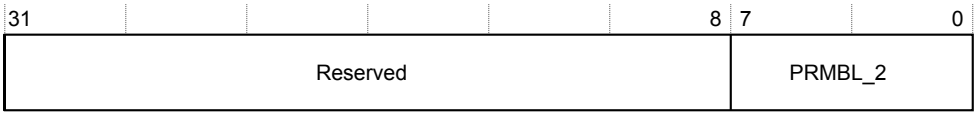

Figure 9-183 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-190 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

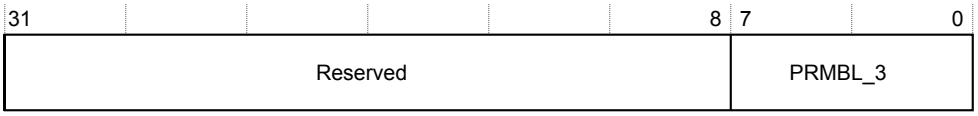

Figure 9-184 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-191 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.9 css600_apbrom_gpr introduction

This section describes the functions and programmers model of the `css600_apbrom_gpr`.

This section contains the following subsections:

9.9.1 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.

Table 9-192 css600_apbrom_gpr - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	ROMEntry0	RO	0x--------	32	*ROM Entries register 0, ROMEntry0* on page 9-314
0x0004	ROMEntry1	RO	0x--------	32	*ROM Entries register 1, ROMEntry1* on page 9-315
0x0008	ROMEntry2	RO	0x--------	32	*ROM Entries register 2, ROMEntry2* on page 9-316
…	…	…	…	…	…
0x07FC	ROMEntry511	RO	0x--------	32	*ROM Entries register 511, ROMEntry511* on page 9-317
0x0A00	DBGPCR0	RW	0x00000001	32	*Debug Power Control Register 0, DBGPCR0* on page 9-318
0x0A04	DBGPCR1	RW	0x00000001	32	*Debug Power Control Register 1, DBGPCR1* on page 9-319
0x0A08	DBGPCR2	RW	0x00000001	32	*Debug Power Control Register 2, DBGPCR2* on page 9-320
…	…	…	…	…	…
0x0A7C	DBGPCR31	RW	0x00000001	32	*Debug Power Control Register 31, DBGPCR31* on page 9-321
0x0A80	DBGPSR0	RW	0x00000000	32	*Debug Power Status Register 0, DBGPSR0* on page 9-322
0x0A84	DBGPSR1	RW	0x00000000	32	*Debug Power Status Register 1, DBGPSR1* on page 9-323
0x0A88	DBGPSR2	RW	0x00000000	32	*Debug Power Status Register 2, DBGPSR2* on page 9-324
…	…	…	…	…	…
0x0AFC	DBGPSR31	RW	0x00000000	32	*Debug Power Status Register 31, DBGPSR31* on page 9-325
0x0B00	SYSPCR0	RW	0x00000001	32	*System Power Control Register 0, SYSPCR0* on page 9-326
0x0B04	SYSPCR1	RW	0x00000001	32	*System Power Control Register 1, SYSPCR1* on page 9-327
0x0B08	SYSPCR2	RW	0x00000001	32	*System Power Control Register 2, SYSPCR2* on page 9-328
…	…	…	…	…	…
0x0B7C	SYSPCR31	RW	0x00000001	32	*System Power Control Register 31, SYSPCR31* on page 9-329
0x0B80	SYSPSR0	RW	0x00000000	32	*System Power Status Register 0, SYSPSR0* on page 9-330
0x0B84	SYSPSR1	RW	0x00000000	32	*System Power Status Register 1, SYSPSR1* on page 9-331
0x0B88	SYSPSR2	RW	0x00000000	32	*System Power Status Register 2, SYSPSR2* on page 9-332

Table 9-192 css600_apbrom_gpr - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
…	…	…	…	…	…
0x0BFC	SYSPSR31	RW	0x00000000	32	*System Power Status Register 31, SYSPSR31* on page 9-333
0x0C00	PRIDR0	RO	0x00000000	32	*Power Request ID Register, PRIDR0* on page 9-334
0x0C10	DBGRSTRR	RW	0x00000000	32	*Debug Reset Request Register, DBGRSTRR* on page 9-335
0x0C14	DBGRSTAR	RW	0x00000000	32	*Debug Reset Acknowledge Register, DBGRSTAR* on page 9-336
0x0C18	SYSRSTRR	RW	0x00000000	32	*System Reset Request Register, SYSRSTRR* on page 9-337
0x0C1C	SYSRSTAR	RW	0x00000000	32	*System Reset Acknowledge Register, SYSRSTAR* on page 9-338
0x0FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-339
0x0FBC	DEVARCH	RO	0x47700af7	32	*Device Architecture Register, DEVARCH* on page 9-341
0x0FC8	DEVID	RO	0x000000-0	32	*Device Configuration Register, DEVID* on page 9-342
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-343
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-344
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-345
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-346
0x0FE0	PIDR0	RO	0x000000--	32	*Peripheral Identification Register 0, PIDR0* on page 9-347
0x0FE4	PIDR1	RO	0x000000--	32	*Peripheral Identification Register 1, PIDR1* on page 9-348
0x0FE8	PIDR2	RO	0x000000--	32	*Peripheral Identification Register 2, PIDR2* on page 9-349
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-350
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-351
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-352
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-353
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-354

9.9.2 Register descriptions

This section describes the `css600_apbrom_gpr` registers.

9.9.1 Register summary on page 9-312 provides cross references to individual registers.

Note

The ROM table has a configuration parameter `TIE_OFF_PRESENT`, which, if set to 1, allows for the selective removal of any entry using the **entry_present** input bus. The **entry_present** bus contains one bit per ROM table entry.

If a given **entry_present[n]** bit is tied HIGH, then the value in ROMEntry<n>.PRESENT[1:0] is taken directly from the value in the `ROM_ENTRY<n>` parameter for that entry.

If the **entry_present[n]** input is tied LOW, then a value of `0x3` in ROMEntry<n>.PRESENT[1:0] is modified to read `0x2` to indicate that the value is not present and is not the last entry.

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.

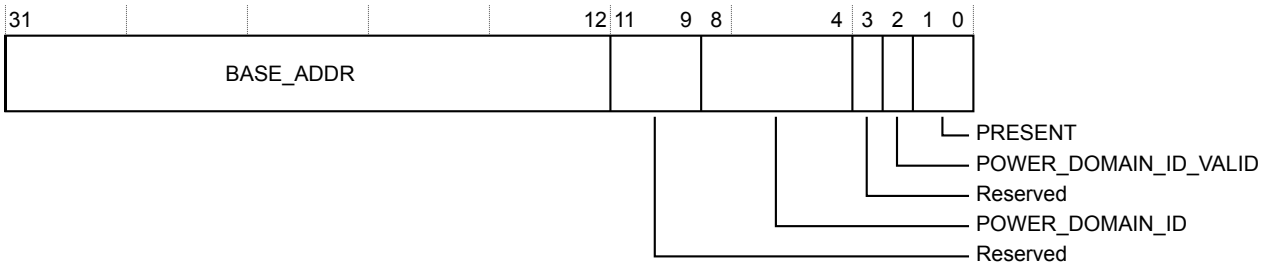

Figure 9-185 ROMEntry0 register bit assignments

The following table shows the bit assignments.

Table 9-193 ROMEntry0 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

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.

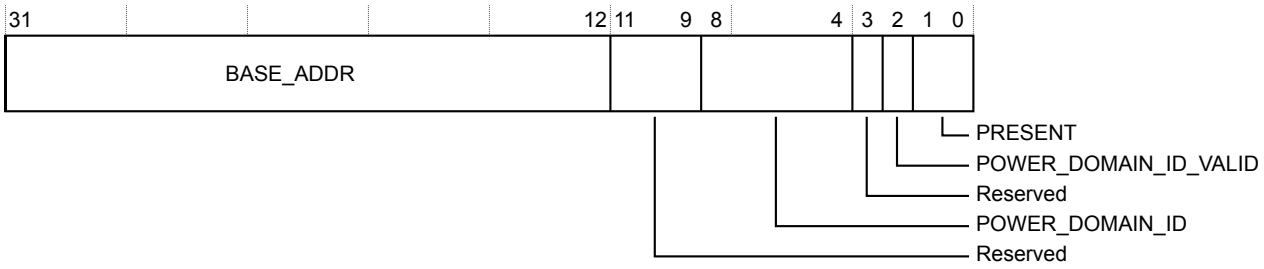

Figure 9-186 ROMEntry1 register bit assignments

The following table shows the bit assignments.

Table 9-194 ROMEntry1 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

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.

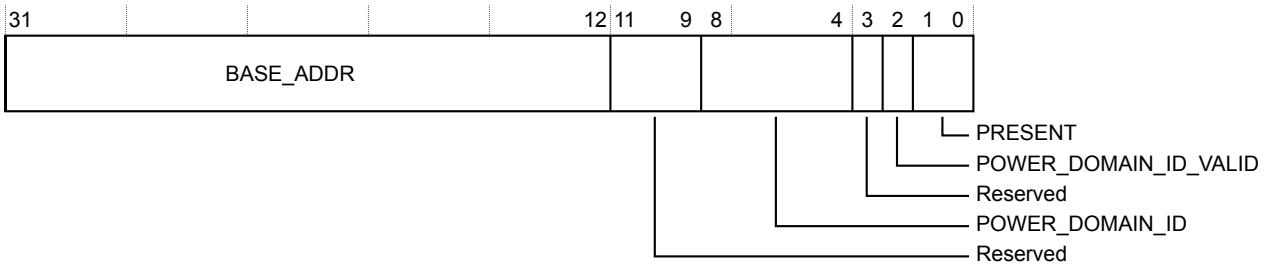

Figure 9-187 ROMEntry2 register bit assignments

The following table shows the bit assignments.

Table 9-195 ROMEntry2 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

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.

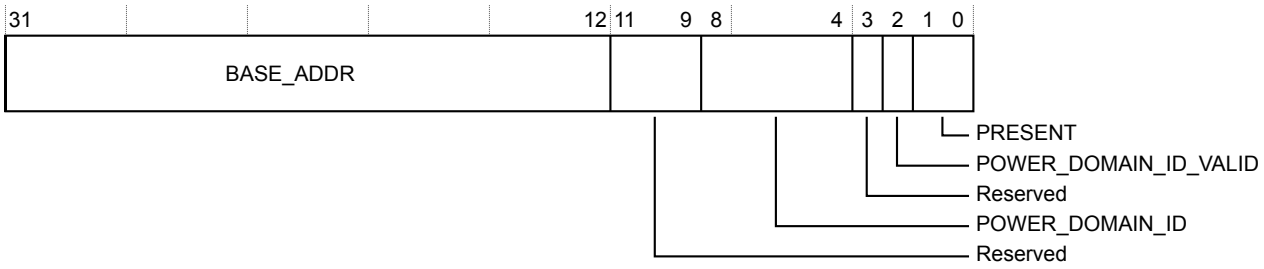

Figure 9-188 ROMEntry511 register bit assignments

The following table shows the bit assignments.

Table 9-196 ROMEntry511 register bit assignments

Bits	Reset value	Name	Function
[31:12]	IMPLEMENTATION_DEFINED	BASE_ADDR	Base address of component.
[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.
[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.

Debug Power Control Register 0, DBGPCR0

Indicates whether power has been requested for a debug domain.

The DBGPCR0 register characteristics are:

Attributes

Offset `0x0a00`
Type Read-write
Reset `0x00000001`
Width 32

The following figure shows the bit assignments.

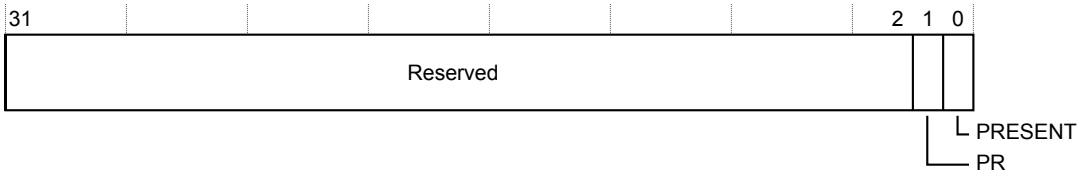

Figure 9-189 DBGPCR0 register bit assignments

The following table shows the bit assignments.

Table 9-197 DBGPCR0 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for debug domain 0. **1** Indicates that power is requested for debug domain0.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for debug domain 0. **0** Indicates that the power request is not implemented for debug domain 0. **1** Indicates that the power request is implemented for debug domain0.

Debug Power Control Register 1, DBGPCR1

Indicates whether power has been requested for a debug domain.

The DBGPCR1 register characteristics are:

Attributes

Offset `0x0a04`
Type Read-write
Reset `0x00000001`
Width 32

The following figure shows the bit assignments.

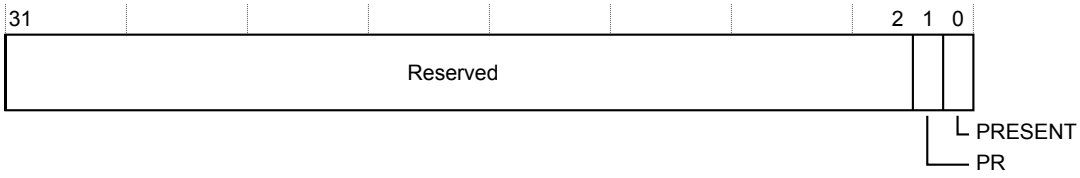

Figure 9-190 DBGPCR1 register bit assignments

The following table shows the bit assignments.

Table 9-198 DBGPCR1 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for debug domain 1. **1** Indicates that power is requested for debug domain1.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for debug domain 1. **0** Indicates that the power request is not implemented for debug domain 1. **1** Indicates that the power request is implemented for debug domain1.

Debug Power Control Register 2, DBGPCR2

Indicates whether power has been requested for a debug domain.

The DBGPCR2 register characteristics are:

Attributes

Offset `0x0a08`
Type Read-write
Reset `0x00000001`
Width 32

The following figure shows the bit assignments.

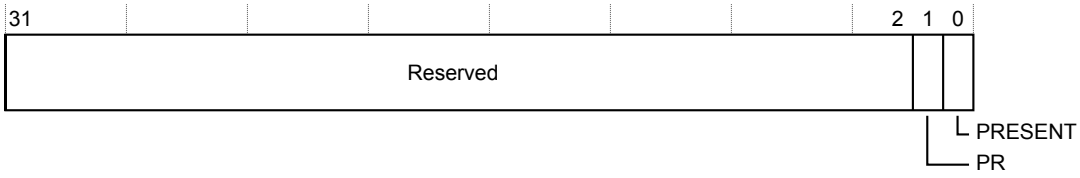

Figure 9-191 DBGPCR2 register bit assignments

The following table shows the bit assignments.

Table 9-199 DBGPCR2 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for debug domain 2. **1** Indicates that power is requested for debug domain2.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for debug domain 2. **0** Indicates that the power request is not implemented for debug domain 2. **1** Indicates that the power request is implemented for debug domain2.

Debug Power Control Register 31, DBGPCR31

Indicates whether power has been requested for a debug domain.

The DBGPCR31 register characteristics are:

Attributes

Offset	`0x0a7c`
Type	Read-write
Reset	`0x00000001`
Width	32

The following figure shows the bit assignments.

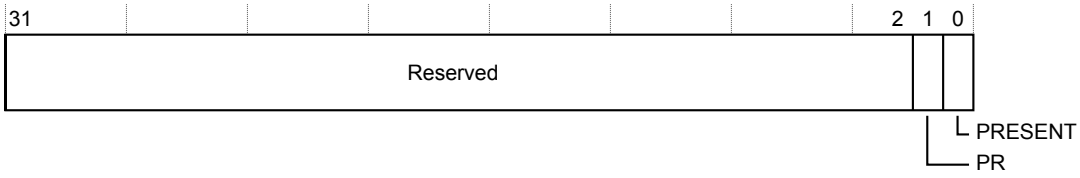

Figure 9-192 DBGPCR31 register bit assignments

The following table shows the bit assignments.

Table 9-200 DBGPCR31 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for debug domain 31. **1** Indicates that power is requested for debug domain31.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for debug domain 31. **0** Indicates that the power request is not implemented for debug domain 31. **1** Indicates that the power request is implemented for debug domain31.

Debug Power Status Register 0, DBGPSR0

Indicates the power status for a debug domain.

The DBGPSR0 register characteristics are:

Attributes

Offset 0x0a80
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

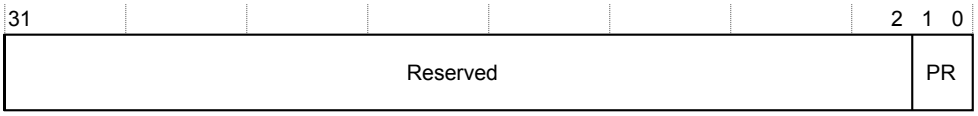

Figure 9-193 DBGPSR0 register bit assignments

The following table shows the bit assignments.

Table 9-201 DBGPSR0 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 Debug domain n might not be powered. 0x1 Debug domain n is powered. 0x2 Reserved. 0x3 Debug domain n is powered and must remain powered until DBGPCRn.PR=0.

Debug Power Status Register 1, DBGPSR1

Indicates the power status for a debug domain.

The DBGPSR1 register characteristics are:

Attributes

Offset 0x0a84
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

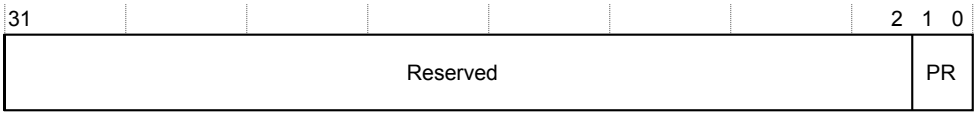

Figure 9-194 DBGPSR1 register bit assignments

The following table shows the bit assignments.

Table 9-202 DBGPSR1 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 Debug domain n might not be powered. 0x1 Debug domain n is powered. 0x2 Reserved. 0x3 Debug domain n is powered and must remain powered until DBGPCRn.PR=0.

Debug Power Status Register 2, DBGPSR2

Indicates the power status for a debug domain.

The DBGPSR2 register characteristics are:

Attributes

Offset 0x0a88
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

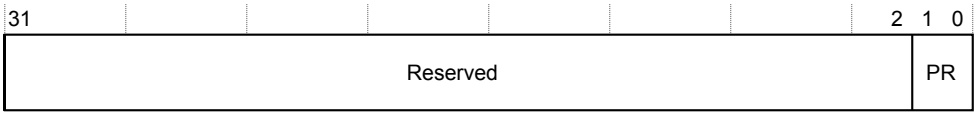

Figure 9-195 DBGPSR2 register bit assignments

The following table shows the bit assignments.

Table 9-203 DBGPSR2 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 Debug domain n might not be powered. 0x1 Debug domain n is powered. 0x2 Reserved. 0x3 Debug domain n is powered and must remain powered until DBGPCRn.PR=0.

Debug Power Status Register 31, DBGPSR31

Indicates the power status for a debug domain.

The DBGPSR31 register characteristics are:

Attributes

Offset 0x0afc
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

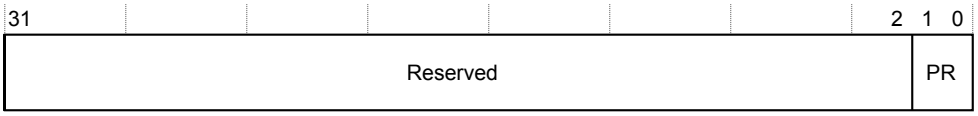

Figure 9-196 DBGPSR31 register bit assignments

The following table shows the bit assignments.

Table 9-204 DBGPSR31 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 Debug domain n might not be powered. 0x1 Debug domain n is powered. 0x2 Reserved. 0x3 Debug domain n is powered and must remain powered until DBGPCRn.PR=0.

System Power Control Register 0, SYSPCR0

Indicates whether power has been requested for a system domain.

The SYSPCR0 register characteristics are:

Attributes

Offset	`0x0b00`
Type	Read-write
Reset	`0x00000001`
Width	32

The following figure shows the bit assignments.

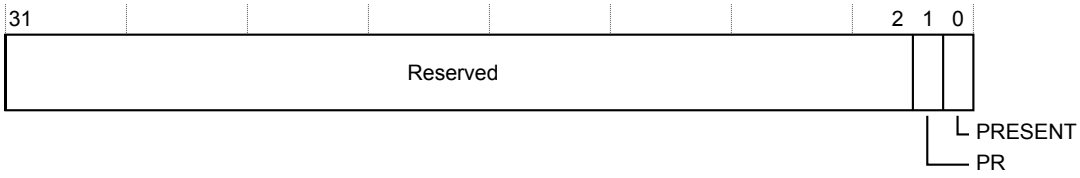

Figure 9-197 SYSPCR0 register bit assignments

The following table shows the bit assignments.

Table 9-205 SYSPCR0 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for system domain 0. **1** Indicates that power is requested for system domain 0.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for system domain 0. **0** Indicates that the power request is not implemented for system domain 0. **1** Indicates that the power request is implemented for system domain 0.

System Power Control Register 1, SYSPCR1

Indicates whether power has been requested for a system domain.

The SYSPCR1 register characteristics are:

Attributes

Offset `0x0b04`
Type Read-write
Reset `0x00000001`
Width 32

The following figure shows the bit assignments.

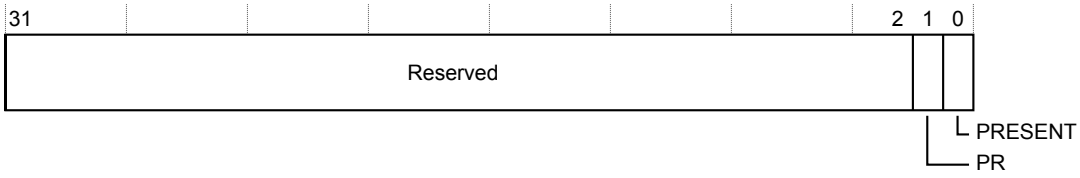

Figure 9-198 SYSPCR1 register bit assignments

The following table shows the bit assignments.

Table 9-206 SYSPCR1 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for system domain 1. **1** Indicates that power is requested for system domain 1.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for system domain 1. **0** Indicates that the power request is not implemented for system domain 1. **1** Indicates that the power request is implemented for system domain 1.

System Power Control Register 2, SYSPCR2

Indicates whether power has been requested for a system domain.

The SYSPCR2 register characteristics are:

Attributes

Offset	`0x0b08`
Type	Read-write
Reset	`0x00000001`
Width	32

The following figure shows the bit assignments.

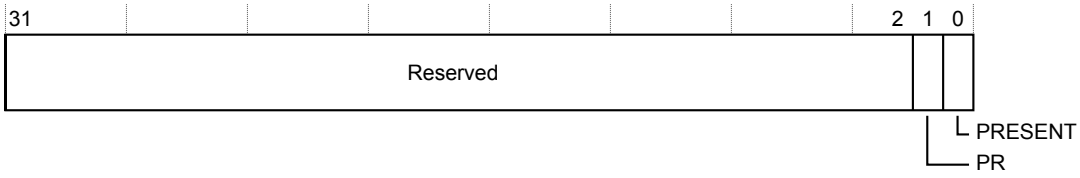

Figure 9-199 SYSPCR2 register bit assignments

The following table shows the bit assignments.

Table 9-207 SYSPCR2 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for system domain 2. **1** Indicates that power is requested for system domain 2.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for system domain 2. **0** Indicates that the power request is not implemented for system domain 2. **1** Indicates that the power request is implemented for system domain 2.

System Power Control Register 31, SYSPCR31

Indicates whether power has been requested for a system domain.

The SYSPCR31 register characteristics are:

Attributes

Offset `0x0b7c`
Type Read-write
Reset `0x00000001`
Width 32

The following figure shows the bit assignments.

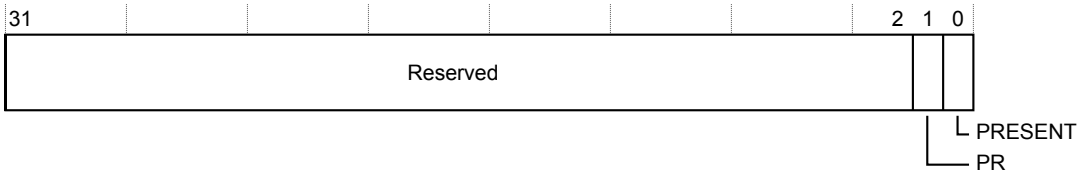

Figure 9-200 SYSPCR31 register bit assignments

The following table shows the bit assignments.

Table 9-208 SYSPCR31 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	PR	Power request. Reserved if DBGPCRn.PRESENT=0 or SYSPCRn=0. **0** Indicates that power is not requested for system domain 31. **1** Indicates that power is requested for system domain 31.
[0]	`0b1`	PRESENT	Indicates the presence of power domain control for system domain 31. **0** Indicates that the power request is not implemented for system domain 31. **1** Indicates that the power request is implemented for system domain 31.

System Power Status Register 0, SYSPSR0

Indicates the power status for a system domain.

The SYSPSR0 register characteristics are:

Attributes

Offset 0x0b80
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

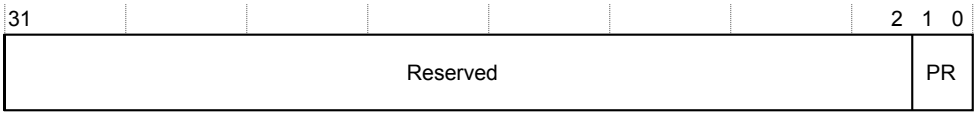

Figure 9-201 SYSPSR0 register bit assignments

The following table shows the bit assignments.

Table 9-209 SYSPSR0 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 System domain n might not be powered. 0x1 System domain n is powered. 0x2 Reserved. 0x3 System domain n is powered and must remain powered until DBGPCRn.PR=0.

System Power Status Register 1, SYSPSR1

Indicates the power status for a system domain.

The SYSPSR1 register characteristics are:

Attributes

Offset 0x0b84
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

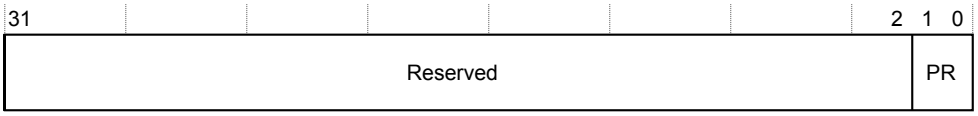

Figure 9-202 SYSPSR1 register bit assignments

The following table shows the bit assignments.

Table 9-210 SYSPSR1 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 System domain n might not be powered. 0x1 System domain n is powered. 0x2 Reserved. 0x3 System domain n is powered and must remain powered until DBGPCRn.PR=0.

System Power Status Register 2, SYSPSR2

Indicates the power status for a system domain.

The SYSPSR2 register characteristics are:

Attributes

Offset 0x0b88
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

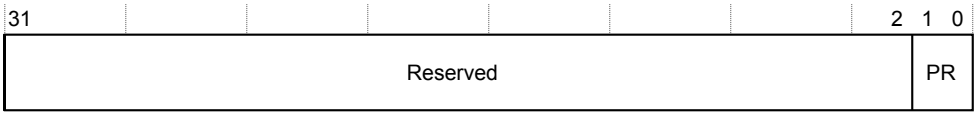

Figure 9-203 SYSPSR2 register bit assignments

The following table shows the bit assignments.

Table 9-211 SYSPSR2 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 System domain n might not be powered. 0x1 System domain n is powered. 0x2 Reserved. 0x3 System domain n is powered and must remain powered until DBGPCRn.PR=0.

System Power Status Register 31, SYSPSR31

Indicates the power status for a system domain.

The SYSPSR31 register characteristics are:

Attributes

Offset 0x0bfc
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

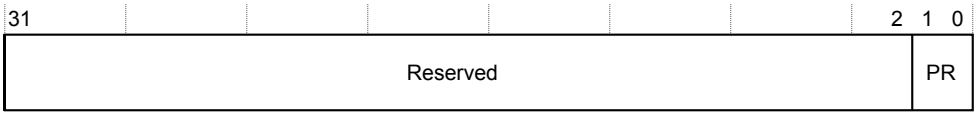

Figure 9-204 SYSPSR31 register bit assignments

The following table shows the bit assignments.

Table 9-212 SYSPSR31 register bit assignments

Bits	Reset value	Name	Function
[1:0]	0b00	PR	Power status. 0x0 System domain n might not be powered. 0x1 System domain n is powered. 0x2 Reserved. 0x3 System domain n is powered and must remain powered until DBGPCRn.PR=0.

Power Request ID Register, PRIDR0

The Power request ID register indicates the version of the power request function.

The PRIDR0 register characteristics are:

Attributes

Offset	`0x0c00`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

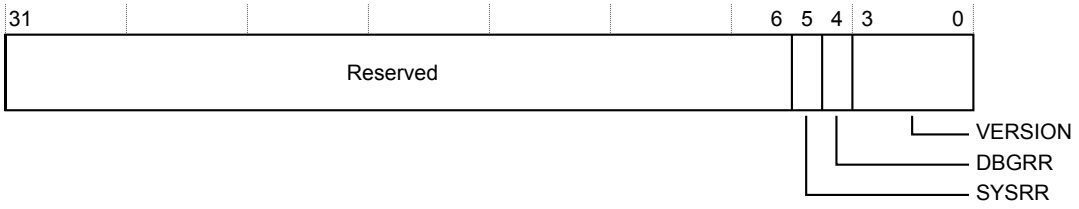

Figure 9-205 PRIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-213 PRIDR0 register bit assignments

Bits	Reset value	Name	Function
[5]	`0b0`	SYSRR	Indicates whether the system reset request functionality is present. **0** System reset request functionality is not implemented. **1** System reset request functionality is implemented. SYSRSTRR and SYSRSTAR are both implemented.
[4]	`0b0`	DBGRR	Indicates whether the debug reset request functionality is present. **0** Debug reset request functionality is not implemented. **1** Debug reset request functionality is implemented. DBGRSTRR and DBGRSTAR are both implemented.
[3:0]	`0b0000`	VERSION	Version of the power request function. Set according to the GPR_PRESENT parameter. `0x0` Power request functionality is not included. `0x1` Power request functionality version 1 is included.

Debug Reset Request Register, DBGRSTRR

Indicates the status of a debug reset request.

The DBGRSTRR register characteristics are:

Attributes

Offset	`0x0c10`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

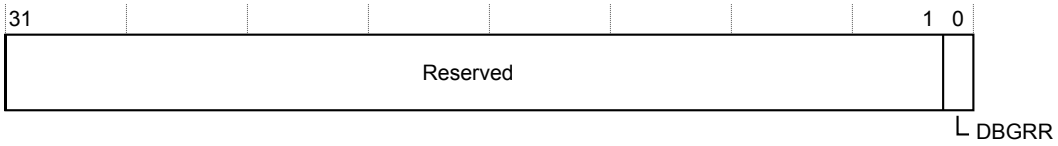

Figure 9-206 DBGRSTRR register bit assignments

The following table shows the bit assignments.

Table 9-214 DBGRSTRR register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	DBGRR	Debug reset request. The software needs to clear this bit after setting it. There is no automatic mechanism to clear it. **0** No reset is requested. **cdbgrstreq** output is LOW. **1** Reset is requested. **cdbgrstreq** output is HIGH.

Debug Reset Acknowledge Register, DBGRSTAR

Acknowledges a debug reset request.

The DBGRSTAR register characteristics are:

Attributes

Offset	`0x0c14`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

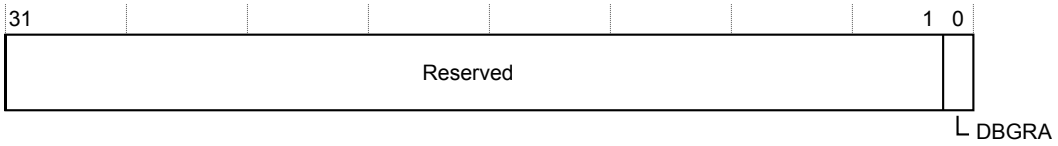

Figure 9-207 DBGRSTAR register bit assignments

The following table shows the bit assignments.

Table 9-215 DBGRSTAR register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	DBGRA	Debug reset acknowledge. **0** No reset is requested or reset is not acknowledged. **1** Reset is acknowledged by the external reset controller.

System Reset Request Register, SYSRSTRR

Indicates the status of a system reset request.

The SYSRSTRR register characteristics are:

Attributes

Offset	`0x0c18`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

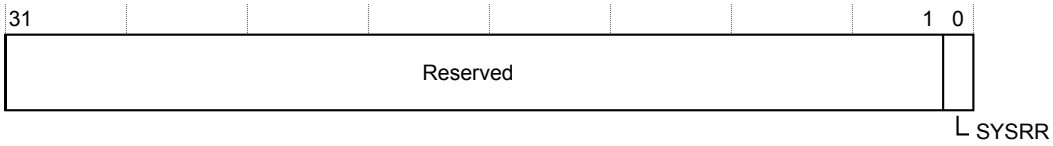

Figure 9-208 SYSRSTRR register bit assignments

The following table shows the bit assignments.

Table 9-216 SYSRSTRR register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	SYSRR	System reset request. The software needs to clear this bit after setting it. There is no automatic mechanism to clear it. **0** No reset is requested. **csysrstreq** output is LOW. **1** Reset is requested. **csysrstreq** output is HIGH.

System Reset Acknowledge Register, SYSRSTAR

Acknowledges a system reset request.

The SYSRSTAR register characteristics are:

Attributes

Offset	`0x0c1c`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

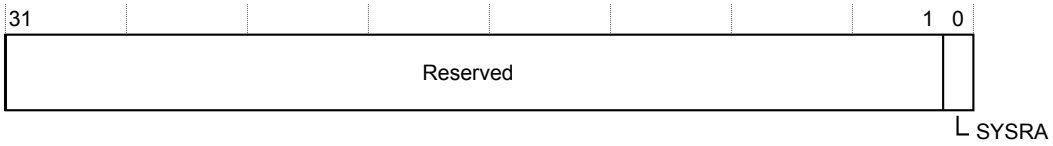

Figure 9-209 SYSRSTAR register bit assignments

The following table shows the bit assignments.

Table 9-217 SYSRSTAR register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	SYSRA	System reset acknowledge. **0** No reset is requested or reset is not acknowledged. **1** Reset is acknowledged by the external reset controller.

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.

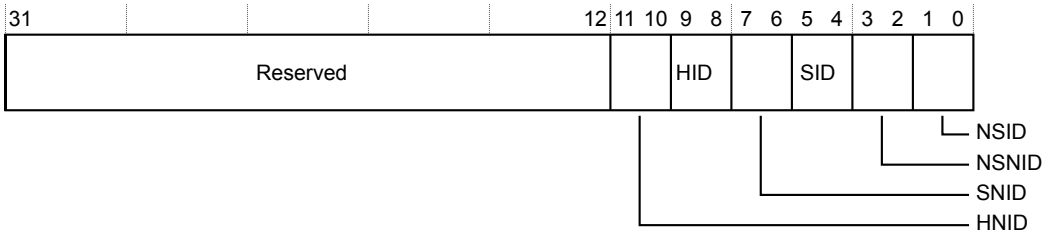

Figure 9-210 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-218 AUTHSTATUS register bit assignments

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.

Table 9-218 AUTHSTATUS register bit assignments (continued)

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.

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.

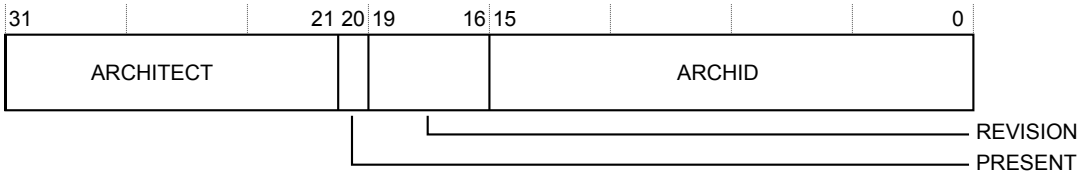

Figure 9-211 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-219 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.

Device Configuration Register, DEVID

This register is IMPLEMENTATION DEFINED for each Part Number and Designer. This 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.

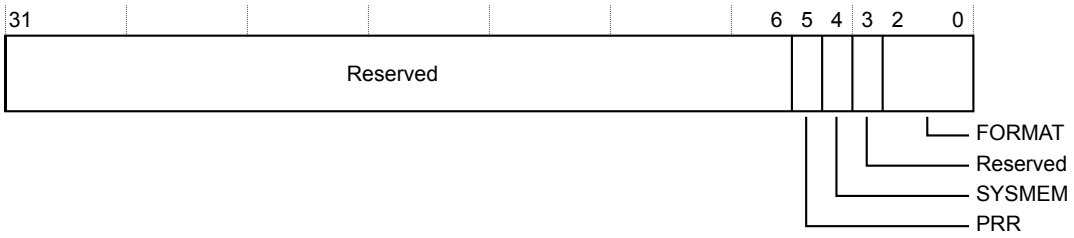

Figure 9-212 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-220 DEVID register bit assignments

Bits	Reset value	Name	Function
[5]	`0b1`	PRR	Indicates that power request functionality is included. Set by the GPR_PRESENT parameter. **0** GPR is not included (css600_rom). **1** GPR is included (css600_rom_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.
[2:0]	`0b000`	FORMAT	Indicates that this is a 32-bit ROM table.

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.

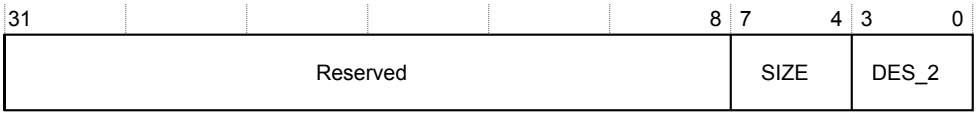

Figure 9-213 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-221 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

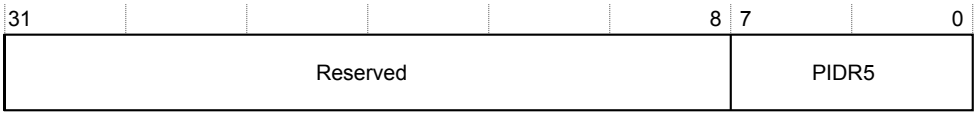

Figure 9-214 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-222 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

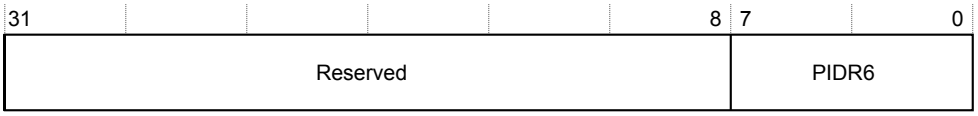

Figure 9-215 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-223 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

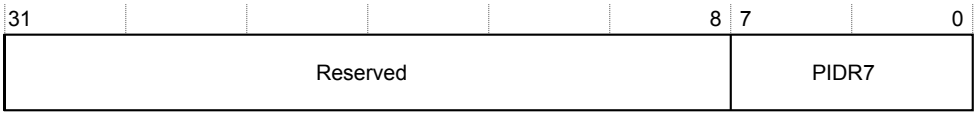

Figure 9-216 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-224 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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.

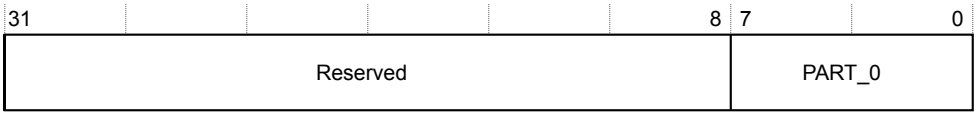

Figure 9-217 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-225 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	IMPLEMENTATION_DEFINED	PART_0	Part number, bits[7:0]. Set by the configuration inputs **part_number[7:0]**

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.

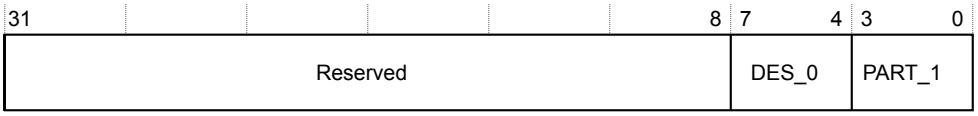

Figure 9-218 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-226 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]**

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.

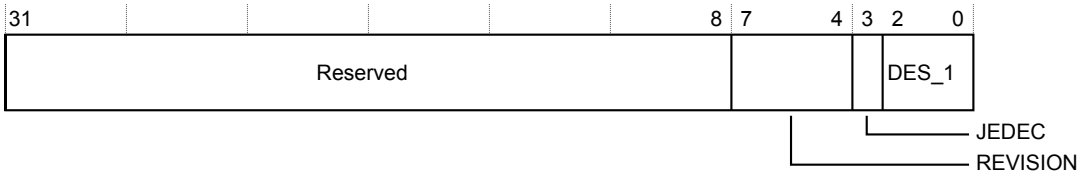

Figure 9-219 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-227 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

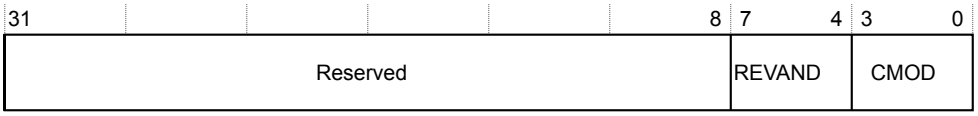

Figure 9-220 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-228 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

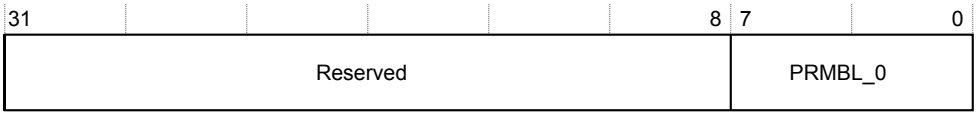

Figure 9-221 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-229 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

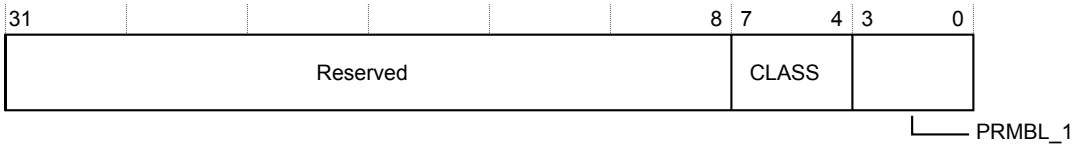

Figure 9-222 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-230 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

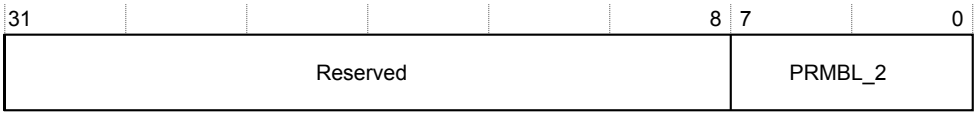

Figure 9-223 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-231 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

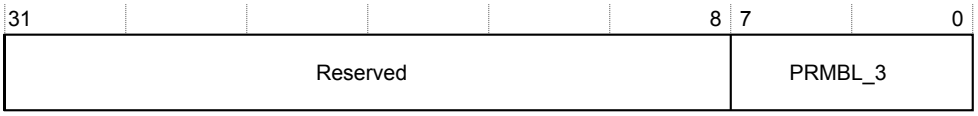

Figure 9-224 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-232 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.10 css600_atbfunnel_prog introduction

This section describes the functions and programmers model of the `css600_atbfunnel_prog`.

This section contains the following subsections:

9.10.1 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.

Table 9-233 css600_atbfunnel_prog - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	FUNNELCONTROL	RW	0x00000300	32	*Funnel Control register, FUNNELCONTROL* on page 9-357
0x0004	PRIORITYCONTROL	RW	0x00000000	32	*Priority Control register, PRIORITYCONTROL* on page 9-360
0x0EEC	ITATBDATA0	RW	0x00000000	32	*Integration test data register, ITATBDATA0* on page 9-361
0x0EF0	ITATBCTR3	RW	0x00000000	32	*Integration test control register 3, ITATBCTR3* on page 9-364
0x0EF4	ITATBCTR2	RW	0x00000000	32	*Integration test control register 2, ITATBCTR2* on page 9-365
0x0EF8	ITATBCTR1	RW	0x00000000	32	*Integration test control register 1, ITATBCTR1* on page 9-366
0x0EFC	ITATBCTR0	RW	0x00000000	32	*Integration test control register 0, ITATBCTR0* on page 9-367
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-368
0x0FA0	CLAIMSET	RW	0x0000000f	32	*Claim Tag Set Register, CLAIMSET* on page 9-369
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-370
0x0FA8	DEVAFF0	RO	0x00000000	32	*Device Affinity register 0, DEVAFF0* on page 9-371
0x0FAC	DEVAFF1	RO	0x00000000	32	*Device Affinity register 1, DEVAFF1* on page 9-372
0x0FB8	AUTHSTATUS	RO	0x00000000	32	*Authentication Status Register, AUTHSTATUS* on page 9-373
0x0FBC	DEVARCH	RO	0x00000000	32	*Device Architecture Register, DEVARCH* on page 9-375
0x0FC0	DEVID2	RO	0x00000000	32	*Device Configuration Register 2, DEVID2* on page 9-376
0x0FC4	DEVID1	RO	0x00000000	32	*Device Configuration Register 1, DEVID1* on page 9-377
0x0FC8	DEVID	RO	0x00000038	32	*Device Configuration Register, DEVID* on page 9-378
0x0FCC	DEVTYPE	RO	0x00000012	32	*Device Type Identifier Register, DEVTYPE* on page 9-379
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-380
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-381
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-382
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-383
0x0FE0	PIDR0	RO	0x000000eb	32	*Peripheral Identification Register 0, PIDR0* on page 9-384

Table 9-233 css600_atbfunnel_prog - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-385
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-386
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-387
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-388
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-389
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-390
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-391

9.10.2 Register descriptions

This section describes the `css600_atbfunnel_prog` registers.

9.10.1 Register summary on page 9-355 provides cross references to individual registers.

Funnel Control register, FUNNELCONTROL

The Funnel Control register is for enabling each of the trace sources and controlling the hold time for switching between them.

The FUNNELCONTROL register characteristics are:

Attributes

Offset	`0x0000`
Type	Read-write
Reset	`0x00000300`
Width	32

The following figure shows the bit assignments.

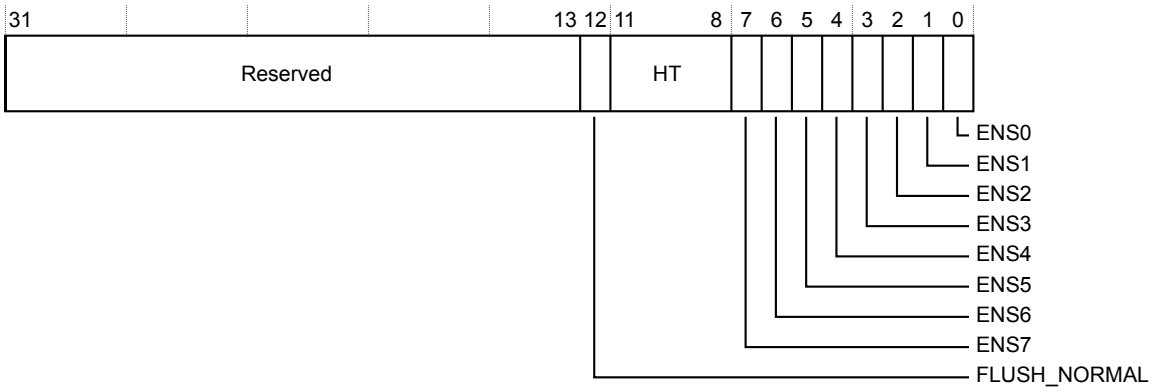

Figure 9-225 FUNNELCONTROL register bit assignments

The following table shows the bit assignments.

Table 9-234 FUNNELCONTROL register bit assignments

Bits	Reset value	Name	Function
[12]	0b0	FLUSH_NORMAL	This bit, when clear, allows slave ports that are already flushed to receive further data even if there are other ports that have not completed flush. If set, a port that has completed flush is not be allowed to receive further data until all ports have completed flush.
[11:8]	0b0011	HT	Hold time. Value sets the minimum hold time before switching trace sources (funnel inputs) based on the ID. Value used is programmed value + 1. 0x0 1 transaction hold time. 0x1 2 transactions hold time. 0x2 3 transactions hold time. 0x3 4 transactions hold time. 0x4 5 transactions hold time. 0x5 6 transactions hold time. 0x6 7 transactions hold time. 0x7 8 transactions hold time. 0x8 9 transactions hold time. 0x9 10 transactions hold time. 0xA 11 transactions hold time. 0xB 12 transactions hold time. 0xC 13 transactions hold time. 0xD 14 transactions hold time. 0xE 15 transactions hold time. 0xF Reserved.
[7]	0b0	ENS7	Enable slave port 7. **0** Slave port disabled. **1** Slave port enabled.
[6]	0b0	ENS6	Enable slave port 6. **0** Slave port disabled. **1** Slave port enabled.
[5]	0b0	ENS5	Enable slave port 5. **0** Slave port disabled. **1** Slave port enabled.
[4]	0b0	ENS4	Enable slave port 4. **0** Slave port disabled. **1** Slave port enabled.
[3]	0b0	ENS3	Enable slave port 3. **0** Slave port disabled. **1** Slave port enabled.

Table 9-234 FUNNELCONTROL register bit assignments (continued)

Bits	Reset value	Name	Function
[2]	0b0	ENS2	Enable slave port 2. **0** Slave port disabled. **1** Slave port enabled.
[1]	0b0	ENS1	Enable slave port 1. **0** Slave port disabled. **1** Slave port enabled.
[0]	0b0	ENS0	Enable slave port 0. **0** Slave port disabled. **1** Slave port enabled.

Priority Control register, PRIORITYCONTROL

The Priority Control register is for setting the priority of each port of the funnel. It is a requirement of the programming software that the ports are all disabled before the priority control register contents are changed. Changing the port priorities in real time is not supported. If the priority control register is written when one or more of the ports are enabled, then the write is silently rejected and the value in the priority control register remains unchanged. The lower the priority value, the higher is its priority when selecting the next port to be serviced. If two or more ports have the same priority value, then the lowest numbered port is serviced first.

The PRIORITYCONTROL register characteristics are:

Attributes

Offset `0x0004`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

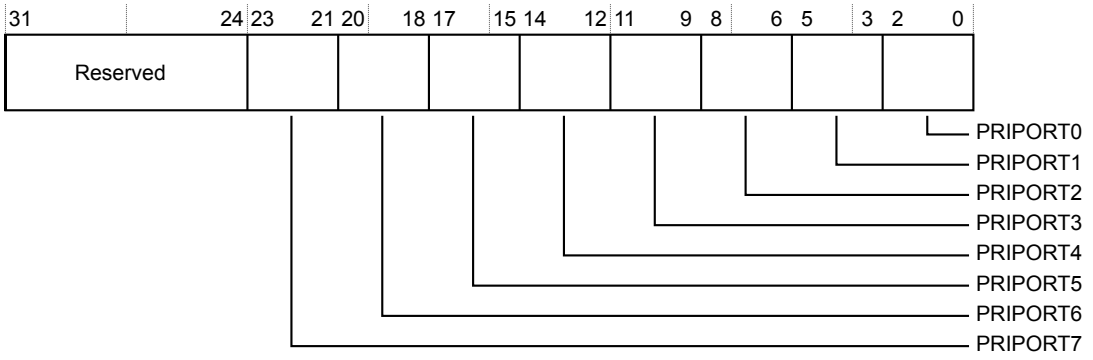

Figure 9-226 PRIORITYCONTROL register bit assignments

The following table shows the bit assignments.

Table 9-235 PRIORITYCONTROL register bit assignments

Bits	Reset value	Name	Function
[23:21]	`0b000`	PRIPORT7	Priority value for port 7
[20:18]	`0b000`	PRIPORT6	Priority value for port 6
[17:15]	`0b000`	PRIPORT5	Priority value for port 5
[14:12]	`0b000`	PRIPORT4	Priority value for port 4
[11:9]	`0b000`	PRIPORT3	Priority value for port 3
[8:6]	`0b000`	PRIPORT2	Priority value for port 2
[5:3]	`0b000`	PRIPORT1	Priority value for port 1
[2:0]	`0b000`	PRIPORT0	Priority value for port 0

Integration test data register, ITATBDATA0

This register allows observability and controllability of the ATDATA buses into and out of the funnel. For slave signals coming into the funnel, the register views the ports that are selected through the funnel control register. Only one port must be selected for integration test.

The ITATBDATA0 register characteristics are:

Attributes

Offset	`0x0eec`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

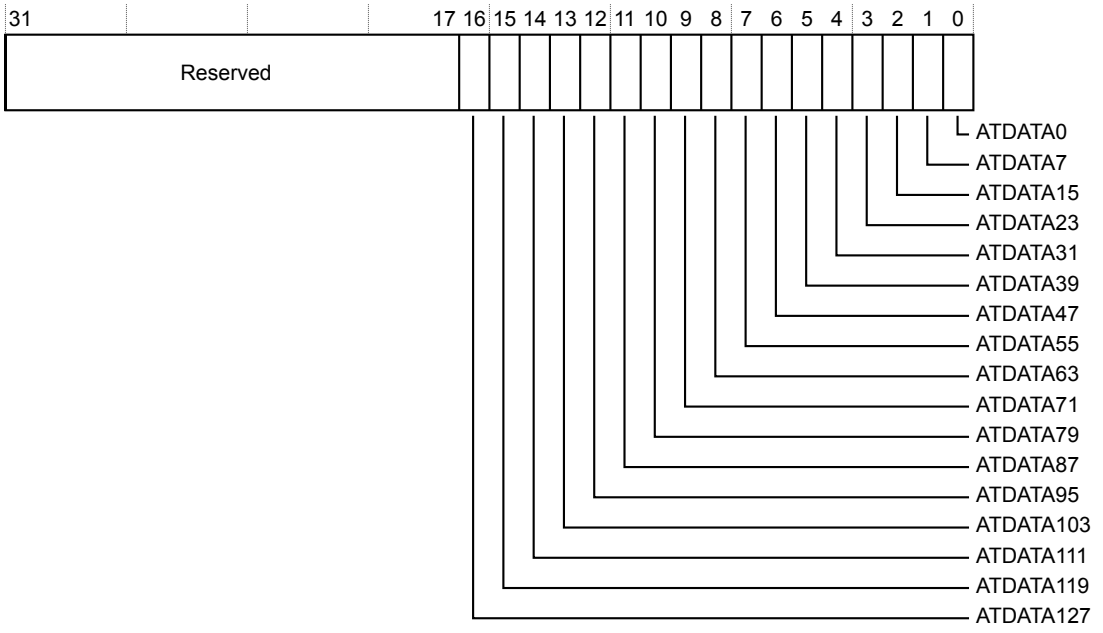

Figure 9-227 ITATBDATA0 register bit assignments

The following table shows the bit assignments.

Table 9-236 ITATBDATA0 register bit assignments

Bits	Reset value	Name	Function
[16]	`0b0`	ATDATA127	Reads **atdata_s[127]** and writes **atdata_m[127]**. **0** On reads, the value of **atdata_s[127]** is 0. On writes, sets **atdata_m[127]** to 0. **1** On reads, the value of **atdata_s[127]** is 1. On writes, sets **atdata_m[127]** to 1.
[15]	`0b0`	ATDATA119	Reads **atdata_s[119]** and writes **atdata_m[119]**. **0** On reads, the value of **atdata_s[119]** is 0. On writes, sets **atdata_m[119]** to 0. **1** On reads, the value of **atdata_s[119]** is 1. On writes, sets **atdata_m[119]** to 1.

Table 9-236 ITATBDATA0 register bit assignments (continued)

Bits	Reset value	Name	Function
[14]	0b0	ATDATA111	Reads **atdata_s[111]** and writes **atdata_m[111]**. **0** On reads, the value of **atdata_s[111]** is 0. On writes, sets **atdata_m[111]** to 0. **1** On reads, the value of **atdata_s[111]** is 1. On writes, sets **atdata_m[111]** to 1.
[13]	0b0	ATDATA103	Reads **atdata_s[103]** and writes **atdata_m[103]**. **0** On reads, the value of **atdata_s[103]** is 0. On writes, sets **atdata_m[103]** to 0. **1** On reads, the value of **atdata_s[103]** is 1. On writes, sets **atdata_m[103]** to 1.
[12]	0b0	ATDATA95	Reads **atdata_s[95]** and writes **atdata_m[95]**. **0** On reads, the value of **atdata_s[95]** is 0. On writes, sets **atdata_m[95]** to 0. **1** On reads, the value of **atdata_s[95]** is 1. On writes, sets **atdata_m[95]** to 1.
[11]	0b0	ATDATA87	Reads **atdata_s[87]** and writes **atdata_m[87]**. **0** On reads, the value of **atdata_s[87]** is 0. On writes, sets **atdata_m[87]** to 0. **1** On reads, the value of **atdata_s[87]** is 1. On writes, sets **atdata_m[87]** to 1.
[10]	0b0	ATDATA79	Reads **atdata_s[79]** and writes **atdata_m[79]**. **0** On reads, the value of **atdata_s[79]** is 0. On writes, sets **atdata_m[79]** to 0. **1** On reads, the value of **atdata_s[79]** is 1. On writes, sets **atdata_m[79]** to 1.
[9]	0b0	ATDATA71	Reads **atdata_s[71]** and writes **atdata_m[71]**. **0** On reads, the value of **atdata_s[71]** is 0. On writes, sets **atdata_m[71]** to 0. **1** On reads, the value of **atdata_s[71]** is 1. On writes, sets **atdata_m[71]** to 1.
[8]	0b0	ATDATA63	Reads **atdata_s[63]** and writes **atdata_m[63]**. **0** On reads, the value of **atdata_s[63]** is 0. On writes, sets **atdata_m[63]** to 0. **1** On reads, the value of **atdata_s[63]** is 1. On writes, sets **atdata_m[63]** to 1.
[7]	0b0	ATDATA55	Reads **atdata_s[55]** and writes **atdata_m[55]**. **0** On reads, the value of **atdata_s[55]** is 0. On writes, sets **atdata_m[55]** to 0. **1** On reads, the value of **atdata_s[55]** is 1. On writes, sets **atdata_m[55]** to 1.
[6]	0b0	ATDATA47	Reads **atdata_s[47]** and writes **atdata_m[47]**. **0** On reads, the value of **atdata_s[47]** is 0. On writes, sets **atdata_m[47]** to 0. **1** On reads, the value of **atdata_s[47]** is 1. On writes, sets **atdata_m[47]** to 1.
[5]	0b0	ATDATA39	Reads **atdata_s[39]** and writes **atdata_m[39]**. **0** On reads, the value of **atdata_s[39]** is 0. On writes, sets **atdata_m[39]** to 0. **1** On reads, the value of **atdata_s[39]** is 1. On writes, sets **atdata_m[39]** to 1.
[4]	0b0	ATDATA31	Reads **atdata_s[31]** and writes **atdata_m[31]**. **0** On reads, the value of **atdata_s[31]** is 0. On writes, sets **atdata_m[31]** to 0. **1** On reads, the value of **atdata_s[31]** is 1. On writes, sets **atdata_m[31]** to 1.

Table 9-236 ITATBDATA0 register bit assignments (continued)

Bits	Reset value	Name	Function
[3]	0b0	ATDATA23	Reads **atdata_s[23]** and writes **atdata_m[23]**. **0** On reads, the value of **atdata_s[23]** is 0. On writes, sets **atdata_m[23]** to 0. **1** On reads, the value of **atdata_s[23]** is 1. On writes, sets **atdata_m[23]** to 1.
[2]	0b0	ATDATA15	Reads **atdata_s[15]** and writes **atdata_m[15]**. **0** On reads, the value of **atdata_s[15]** is 0. On writes, sets **atdata_m[15]** to 0. **1** On reads, the value of **atdata_s[15]** is 1. On writes, sets **atdata_m[15]** to 1.
[1]	0b0	ATDATA7	Reads **atdata_s[7]** and writes **atdata_m[7]**. **0** On reads, the value of **atdata_s[7]** is 0. On writes, sets **atdata_m[7]** to 0. **1** On reads, the value of **atdata_s[7]** is 1. On writes, sets **atdata_m[7]** to 1.
[0]	0b0	ATDATA0	Reads **atdata_s[0]** and writes **atdata_m[0]**. **0** On reads, the value of **atdata_s[0]** is 0. On writes, sets **atdata_m[0]** to 0. **1** On reads, the value of **atdata_s[0]** is 1. On writes, sets **atdata_m[0]** to 1.

Integration test control register 3, ITATBCTR3

This register allows observability and controllability of the SYNCREQ signals into, and out of, the funnel. Only one slave port must be selected for integration test. The syncreq receiver on the master port has a latching function to capture a pulse arriving on that input. The arrival of a pulse sets the latch so that the value can be read. Reading the register clears the latch. Reading a 1 indicates that a **syncreq_m** pulse arrived since the last read. Reading a 0 indicates that no **syncreq_m** pulse has arrived. Writing a 1 to the register causes a **syncreq_s** pulse to be generated to the upstream component.

The ITATBCTR3 register characteristics are:

Attributes

Offset	`0x0ef0`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

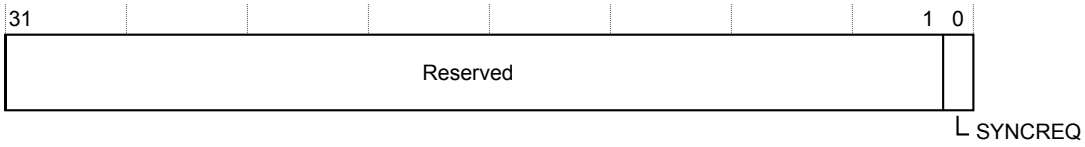

Figure 9-228 ITATBCTR3 register bit assignments

The following table shows the bit assignments.

Table 9-237 ITATBCTR3 register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	SYNCREQ	Reads and controls the SYNCREQ signals into, and out of, the funnel. Reading clears the latch. **0** On reads: no **syncreq_m** pulse has arrived. On writes: no effect. **1** On reads: a **syncreq_m** pulse arrived since the last read. On writes: generates a **syncreq_s** pulse to the upstream component.

Integration test control register 2, ITATBCTR2

This register allows observability and controllability of the afvalid and atready signals into, and out of, the funnel. For slave signals coming into the funnel, the register views the ports that are selected through the funnel control register. Only one port must be selected for integration test.

The ITATBCTR2 register characteristics are:

Attributes

Offset	`0x0ef4`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

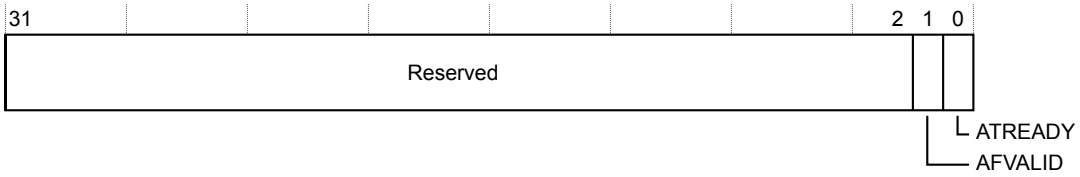

Figure 9-229 ITATBCTR2 register bit assignments

The following table shows the bit assignments.

Table 9-238 ITATBCTR2 register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	AFVALID	Reads and controls the afvalid signals into, and out of, the funnel. **0** On reads: **afvalid_m** is LOW. On writes: sets **afvalid_s** LOW. **1** On reads: **afvalid_m** is HIGH. On writes: sets **afvalid_s** HIGH.
[0]	`0b0`	ATREADY	Reads and controls the atready signal into, and out of, the funnel. **0** On reads: **atready_m** is LOW. On writes: sets **atready_s** LOW. **1** On reads: **atready_m** is HIGH. On writes: sets **atready_s** HIGH.

Integration test control register 1, ITATBCTR1

This register allows observability and controllability of the ATID buses into, and out of, the funnel. For slave signals coming into the funnel, the register views the ports that are selected through the funnel control register. Only one port must be selected for integration test.

The ITATBCTR1 register characteristics are:

Attributes

Offset `0x0ef8`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

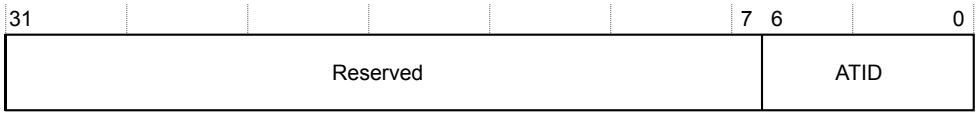

Figure 9-230 ITATBCTR1 register bit assignments

The following table shows the bit assignments.

Table 9-239 ITATBCTR1 register bit assignments

Bits	Reset value	Name	Function
[6:0]	`0b0000000`	ATID	When read returns the value on **atid_s**, when written drives the value on **atid_m**.

Integration test control register 0, ITATBCTR0

This register allows observability and controllability of the ATBYTES buses, and AFREADY and ATVALID signals into, and out of, the funnel. For slave signals coming into the funnel, the register views the ports that are selected through the funnel control register. Only one port must be selected for integration test.

The ITATBCTR0 register characteristics are:

Attributes

Offset	`0x0efc`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

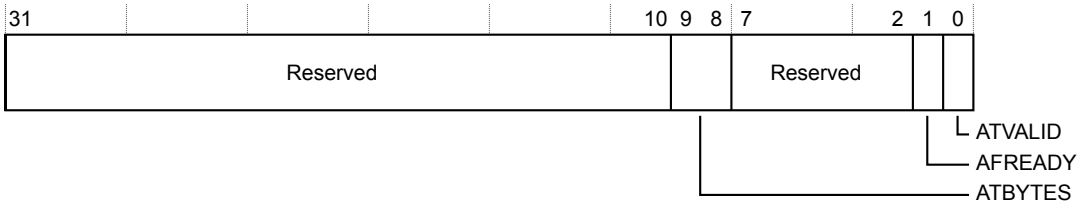

Figure 9-231 ITATBCTR0 register bit assignments

The following table shows the bit assignments.

Table 9-240 ITATBCTR0 register bit assignments

Bits	Reset value	Name	Function
[9:8]	`0b00`	ATBYTES	Reads the value on **atbytes_s[1:0]** and writes the values on **atbytes_m[1:0]**.
[1]	`0b0`	AFREADY	Reads and controls the afready signals into, and out of, the funnel. **0** On reads: **afready_s** is LOW. On writes: sets **afready_m** LOW. **1** On reads: **afready_s** is HIGH. On writes: sets **afready_m** HIGH.
[0]	`0b0`	ATVALID	Reads and controls the atvalid signals into, and out of, the funnel. **0** On reads: **atvalid_s** is LOW. On writes: sets **atvalid_m** LOW. **1** On reads: **atvalid_s** is HIGH. On writes: sets **atvalid_m** HIGH.

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.

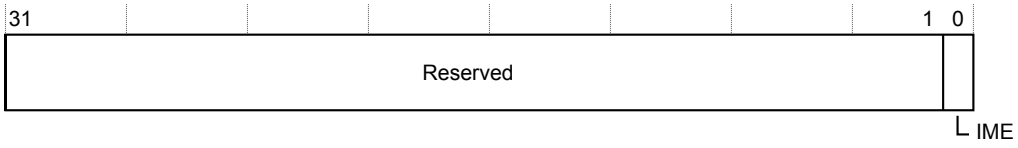

Figure 9-232 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-241 ITCTRL register bit assignments

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

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 `0x0000000f`

Width 32

The following figure shows the bit assignments.

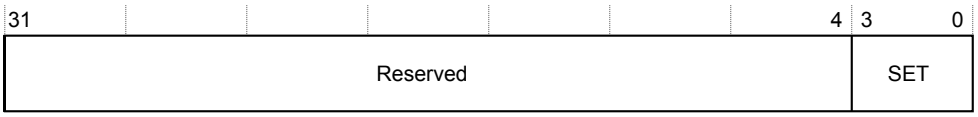

Figure 9-233 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-242 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	SET	A bit programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

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.

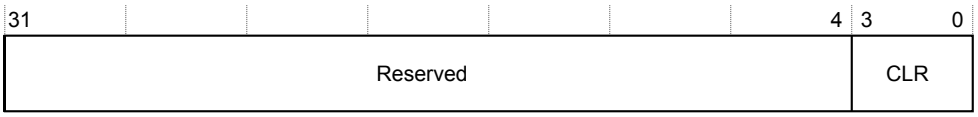

Figure 9-234 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-243 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	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.

Device Affinity register 0, DEVAFF0

Enables a debugger to determine if two components have an affinity with each other.

The DEVAFF0 register characteristics are:

Attributes

Offset	`0x0fa8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 9-235 DEVAFF0 register bit assignments

The following table shows the bit assignments.

Table 9-244 DEVAFF0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVAFF0	This field is RAZ.

Device Affinity register 1, DEVAFF1

Enables a debugger to determine if two components have an affinity with each other.

The DEVAFF1 register characteristics are:

Attributes

Offset	`0x0fac`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

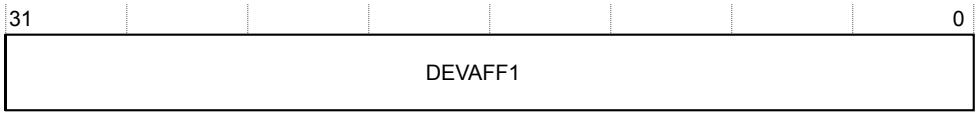

Figure 9-236 DEVAFF1 register bit assignments

The following table shows the bit assignments.

Table 9-245 DEVAFF1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVAFF1	This field is RAZ.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

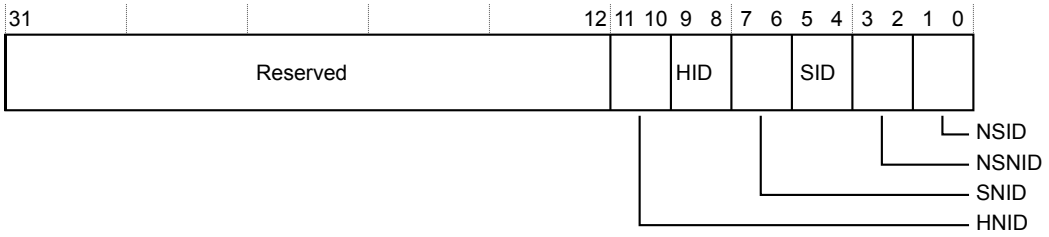

Figure 9-237 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-246 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	`0b00`	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-246 AUTHSTATUS register bit assignments (continued)

Bits	Reset value	Name	Function
[3:2]	0b00	NSNID	Non-secure non-invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.
[1:0]	0b00	NSID	Non-secure invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

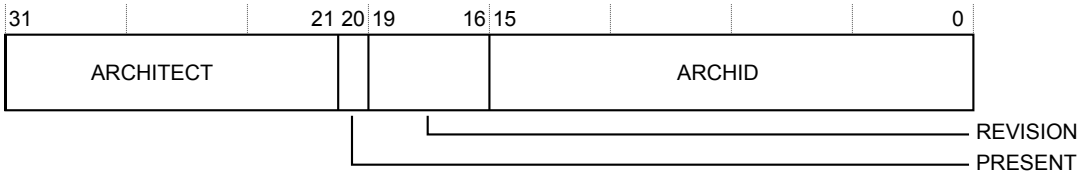

Figure 9-238 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-247 DEVARCH register bit assignments

Bits	Reset value	Name	Function
[31:21]	`0b00000000000`	ARCHITECT	Returns 0.
[20]	`0b0`	PRESENT	Returns 0, indicating that the DEVARCH register is not present.
[19:16]	`0b0000`	REVISION	Returns 0
[15:0]	`0x0`	ARCHID	Returns 0.

Device Configuration Register 2, DEVID2

Contains an IMPLEMENTATION DEFINED value.

The DEVID2 register characteristics are:

Attributes

Offset	`0x0fc0`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

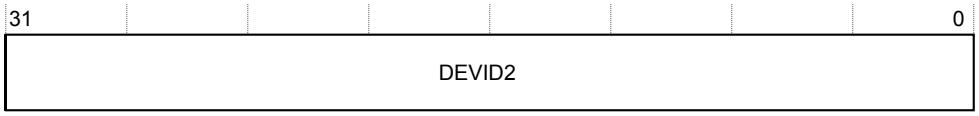

Figure 9-239 DEVID2 register bit assignments

The following table shows the bit assignments.

Table 9-248 DEVID2 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVID2	This field is RAZ.

Device Configuration Register 1, DEVID1

Contains an IMPLEMENTATION DEFINED value.

The DEVID1 register characteristics are:

Attributes

Offset	`0x0fc4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

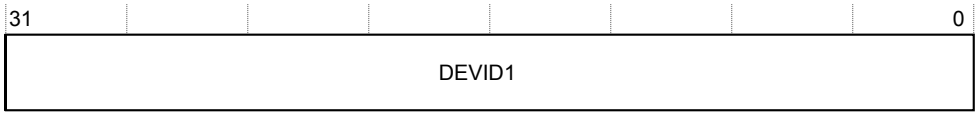

Figure 9-240 DEVID1 register bit assignments

The following table shows the bit assignments.

Table 9-249 DEVID1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVID1	This field is RAZ.

Device Configuration Register, DEVID

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

The DEVID register characteristics are:

Attributes

Offset `0x0fc8`
Type Read-only
Reset `0x00000038`
Width 32

The following figure shows the bit assignments.

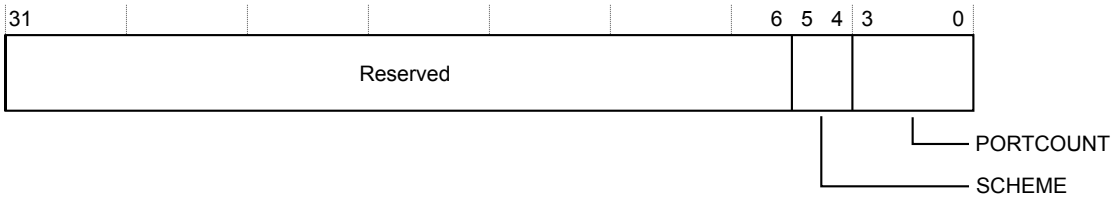

Figure 9-241 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-250 DEVID register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0011`	SCHEME	Indicates priority scheme implemented.
[3:0]	`0b1000`	PORTCOUNT	Indicates the number of input ports connected.

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 `0x00000012`
Width 32

The following figure shows the bit assignments.

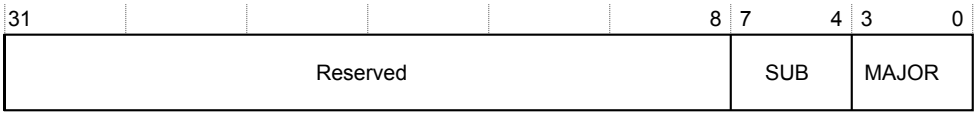

Figure 9-242 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-251 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0001`	SUB	Minor classification. Returns `0x1`, indicating this component is a Funnel/Router.
[3:0]	`0b0010`	MAJOR	Major classification. Returns `0x2`, indicating this component is a Trace Link.

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.

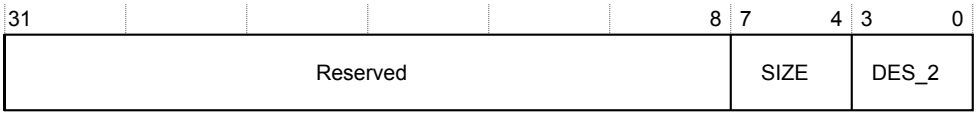

Figure 9-243 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-252 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

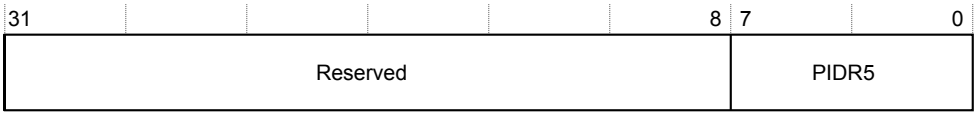

Figure 9-244 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-253 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

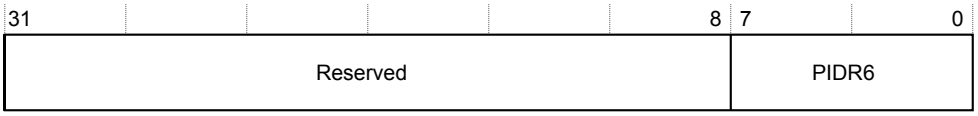

Figure 9-245 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-254 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

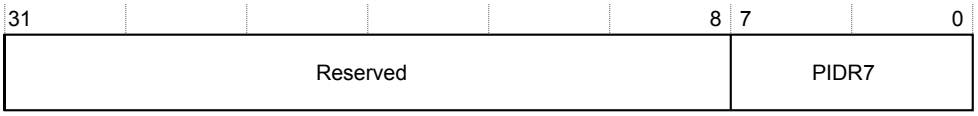

Figure 9-246 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-255 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000eb`

Width 32

The following figure shows the bit assignments.

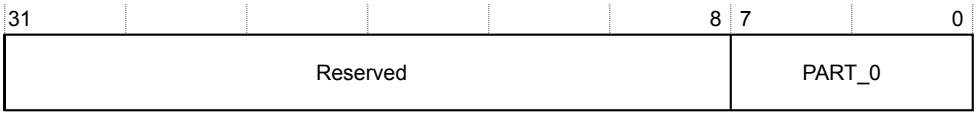

Figure 9-247 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-256 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11101011`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

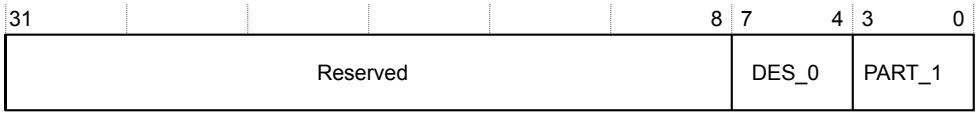

Figure 9-248 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-257 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x0000000b`
Width 32

The following figure shows the bit assignments.

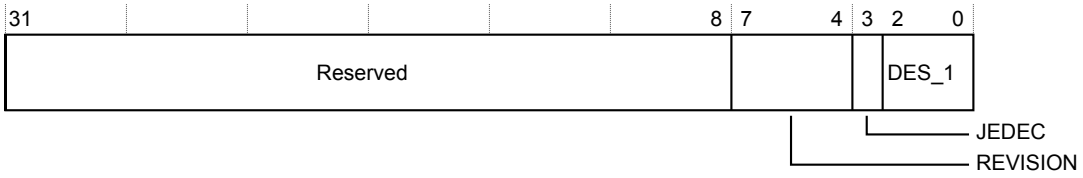

Figure 9-249 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-258 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

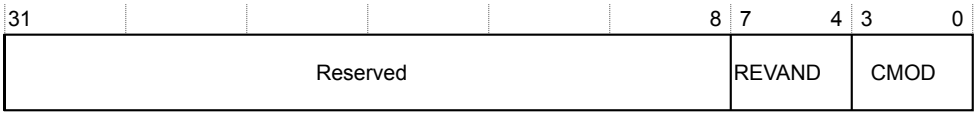

Figure 9-250 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-259 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

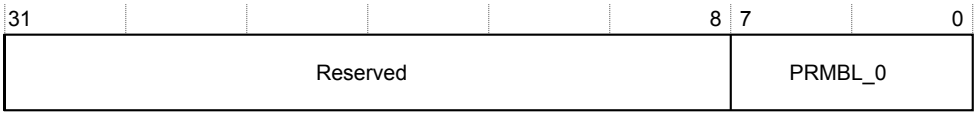

Figure 9-251 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-260 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

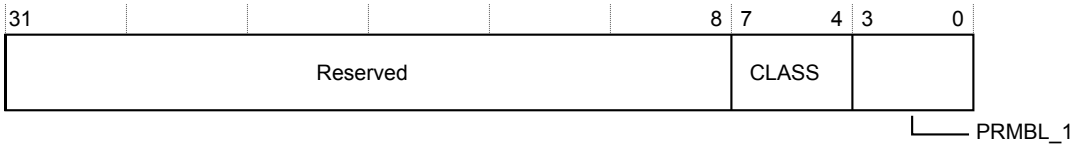

Figure 9-252 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-261 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

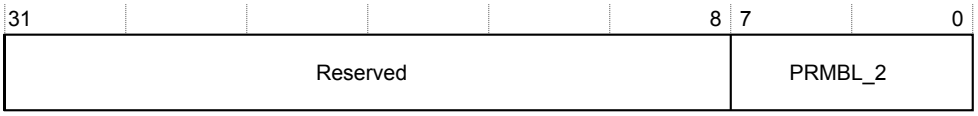

Figure 9-253 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-262 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

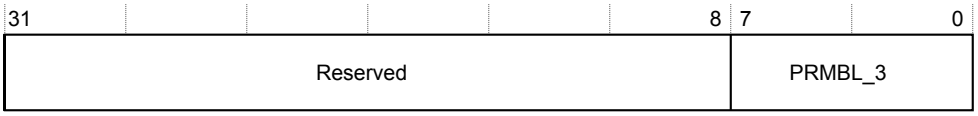

Figure 9-254 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-263 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.11 css600_atbreplicator_prog introduction

This section describes the functions and programmers model of the `css600_atbreplicator_prog`.

This section contains the following subsections:

9.11.1 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.

Table 9-264 css600_atbreplicator_prog - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	IDFILT0	RW	0x00000000	32	*ID filtering control 0 register, IDFILT0* on page 9-394
0x0004	IDFILT1	RW	0x00000000	32	*ID filtering control 1 register, IDFILT1* on page 9-396
0x0EF8	ITATBCTRL	RW	0x00000000	32	*Integration Test Control register, ITATBCTRL* on page 9-398
0x0EFC	ITATBSTAT	RW	0x00000000	32	*Integration Test Status register, ITATBSTAT* on page 9-399
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-400
0x0FA0	CLAIMSET	RW	0x0000000f	32	*Claim Tag Set Register, CLAIMSET* on page 9-401
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-402
0x0FA8	DEVAFF0	RO	0x00000000	32	*Device Affinity register 0, DEVAFF0* on page 9-403
0x0FAC	DEVAFF1	RO	0x00000000	32	*Device Affinity register 1, DEVAFF1* on page 9-404
0x0FB8	AUTHSTATUS	RO	0x00000000	32	*Authentication Status Register, AUTHSTATUS* on page 9-405
0x0FBC	DEVARCH	RO	0x00000000	32	*Device Architecture Register, DEVARCH* on page 9-407
0x0FC0	DEVID2	RO	0x00000000	32	*Device Configuration Register 2, DEVID2* on page 9-408
0x0FC4	DEVID1	RO	0x00000000	32	*Device Configuration Register 1, DEVID1* on page 9-409
0x0FC8	DEVID	RO	0x00000032	32	*Device Configuration Register, DEVID* on page 9-410
0x0FCC	DEVTYPE	RO	0x00000022	32	*Device Type Identifier Register, DEVTYPE* on page 9-411
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-412
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-413
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-414
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-415
0x0FE0	PIDR0	RO	0x000000ec	32	*Peripheral Identification Register 0, PIDR0* on page 9-416
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-417
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-418
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-419

Table 9-264 css600_atbreplicator_prog - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-420
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-421
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-422
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-423

9.11.2 Register descriptions

This section describes the `css600_atbreplicator_prog` registers.

9.11.1 Register summary on page 9-392 provides cross references to individual registers.

ID filtering control 0 register, IDFILT0

Controls ID filtering for master port 0.

The IDFILT0 register characteristics are:

Attributes

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

The following figure shows the bit assignments.

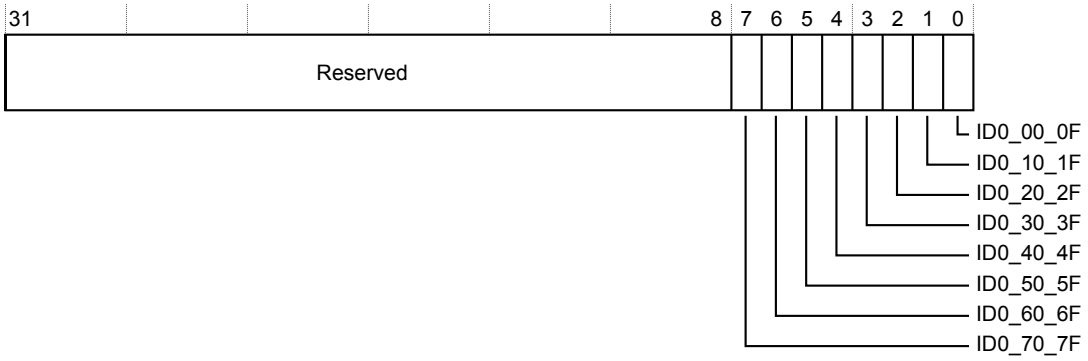

Figure 9-255 IDFILT0 register bit assignments

The following table shows the bit assignments.

Table 9-265 IDFILT0 register bit assignments

Bits	Reset value	Name	Function
[7]	`0b0`	ID0_70_7F	Enable/disable ID filtering for IDs `0x70` to `0x7F`. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.
[6]	`0b0`	ID0_60_6F	Enable/disable ID filtering for IDs `0x60` to `0x6F`. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.
[5]	`0b0`	ID0_50_5F	Enable/disable ID filtering for IDs `0x50` to `0x5F`. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.
[4]	`0b0`	ID0_40_4F	Enable/disable ID filtering for IDs `0x40` to `0x4F`. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.
[3]	`0b0`	ID0_30_3F	Enable/disable ID filtering for IDs `0x30` to `0x3F`. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.

Table 9-265 IDFILT0 register bit assignments (continued)

Bits	Reset value	Name	Function
[2]	0b0	ID0_20_2F	Enable/disable ID filtering for IDs 0x20 to 0x2F. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.
[1]	0b0	ID0_10_1F	Enable/disable ID filtering for IDs 0x10 to 0x1F. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.
[0]	0b0	ID0_00_0F	Enable/disable ID filtering for IDs 0x00 to 0x0F. **0** Transactions with these IDs are passed on to master port 0. **1** Transactions with these IDs are discarded by the replicator.

ID filtering control 1 register, IDFILT1

Controls ID filtering for master port 1.

The IDFILT1 register characteristics are:

Attributes

Offset	`0x0004`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

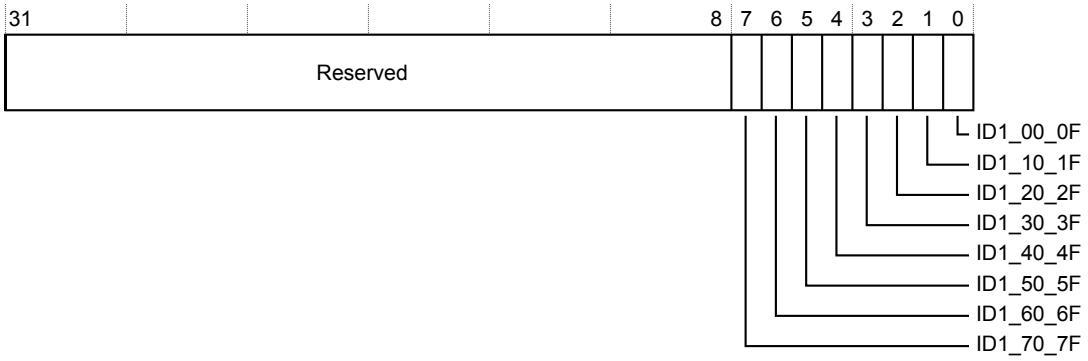

Figure 9-256 IDFILT1 register bit assignments

The following table shows the bit assignments.

Table 9-266 IDFILT1 register bit assignments

Bits	Reset value	Name	Function
[7]	`0b0`	ID1_70_7F	Enable/disable ID filtering for IDs `0x70` to `0x7F`. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.
[6]	`0b0`	ID1_60_6F	Enable/disable ID filtering for IDs `0x60` to `0x6F`. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.
[5]	`0b0`	ID1_50_5F	Enable/disable ID filtering for IDs `0x50` to `0x5F`. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.
[4]	`0b0`	ID1_40_4F	Enable/disable ID filtering for IDs `0x40` to `0x4F`. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.
[3]	`0b0`	ID1_30_3F	Enable/disable ID filtering for IDs `0x30` to `0x3F`. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.

Table 9-266 IDFILT1 register bit assignments (continued)

Bits	Reset value	Name	Function
[2]	0b0	ID1_20_2F	Enable/disable ID filtering for IDs 0x20 to 0x2F. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.
[1]	0b0	ID1_10_1F	Enable/disable ID filtering for IDs 0x10 to 0x1F. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.
[0]	0b0	ID1_00_0F	Enable/disable ID filtering for IDs 0x00 to 0x0F. **0** Transactions with these IDs are passed on to master port 1. **1** Transactions with these IDs are discarded by the replicator.

Integration Test Control register, ITATBCTRL

Integration test control 0. Control of **atready_s** and **atvalid_m**.

The ITATBCTRL register characteristics are:

Attributes

Offset	`0x0ef8`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

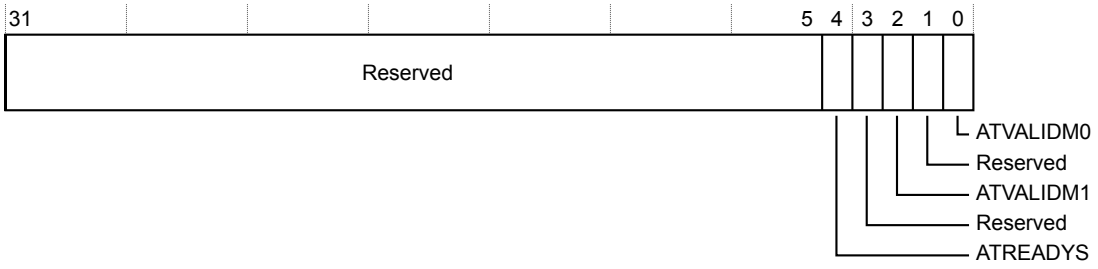

Figure 9-257 ITATBCTRL register bit assignments

The following table shows the bit assignments.

Table 9-267 ITATBCTRL register bit assignments

Bits	Reset value	Name	Function
[4]	`0b0`	ATREADYS	On reads: returns the value written to the register. On writes: **0** Sets static 0 on **atready_s**. **1** Sets static 1 on **atready_s**.
[2]	`0b0`	ATVALIDM1	On reads: returns the value written to the register. On writes: **0** Sets static 0 on atvalid_m1. **1** Sets static 1 on atvalid_m1.
[0]	`0b0`	ATVALIDM0	On reads: returns the value written to the register. On writes: **0** Sets static 0 on atvalid_m0. **1** Sets static 1 on atvalid_m0.

Integration Test Status register, ITATBSTAT

Integration test mode Status. Observability of **atvalid_s** and **atready_m**.

The ITATBSTAT register characteristics are:

Attributes

Offset	`0x0efc`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

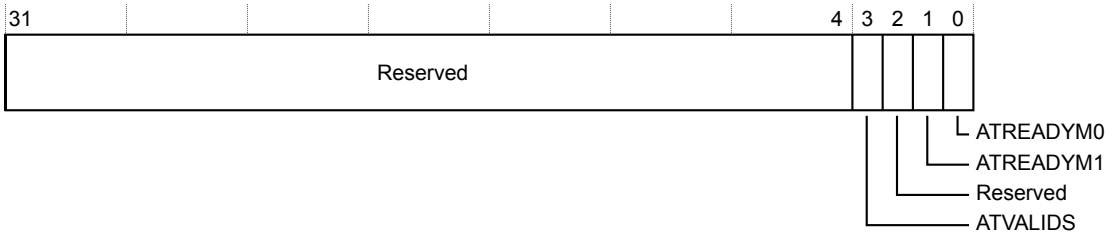

Figure 9-258 ITATBSTAT register bit assignments

The following table shows the bit assignments.

Table 9-268 ITATBSTAT register bit assignments

Bits	Reset value	Name	Function
[3]	`0b0`	ATVALIDS	Returns the value on **atvalid_s**.
[1]	`0b0`	ATREADYM1	Returns the value on atready_m1.
[0]	`0b0`	ATREADYM0	Returns the value on atready_m0.

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.

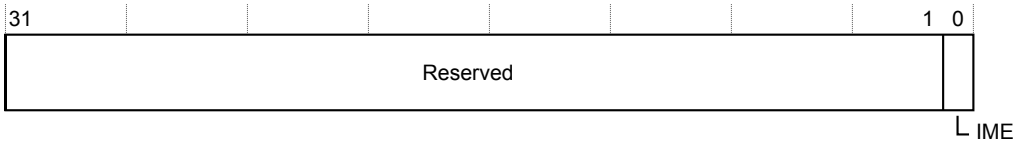

Figure 9-259 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-269 ITCTRL register bit assignments

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

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 `0x0000000f`

Width 32

The following figure shows the bit assignments.

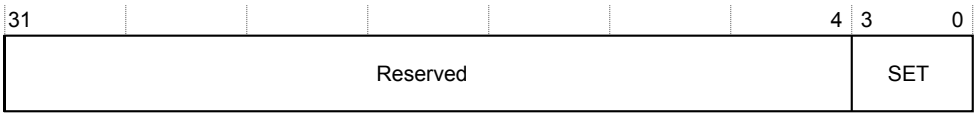

Figure 9-260 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-270 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	SET	A bit programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

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.

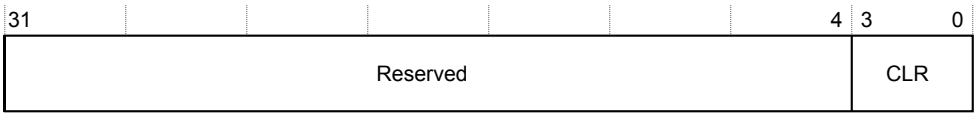

Figure 9-261 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-271 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	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.

Device Affinity register 0, DEVAFF0

Enables a debugger to determine if two components have an affinity with each other.

The DEVAFF0 register characteristics are:

Attributes

Offset `0x0fa8`
Type Read-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

Figure 9-262 DEVAFF0 register bit assignments

The following table shows the bit assignments.

Table 9-272 DEVAFF0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVAFF0	This field is RAZ.

Device Affinity register 1, DEVAFF1

Enables a debugger to determine if two components have an affinity with each other.

The DEVAFF1 register characteristics are:

Attributes

Offset `0x0fac`

Type Read-only

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

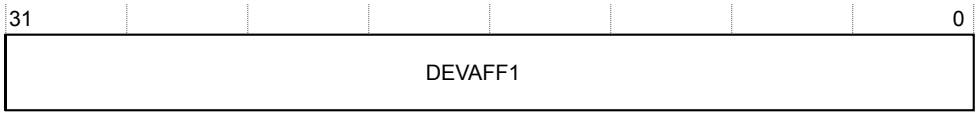

Figure 9-263 DEVAFF1 register bit assignments

The following table shows the bit assignments.

Table 9-273 DEVAFF1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVAFF1	This field is RAZ.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

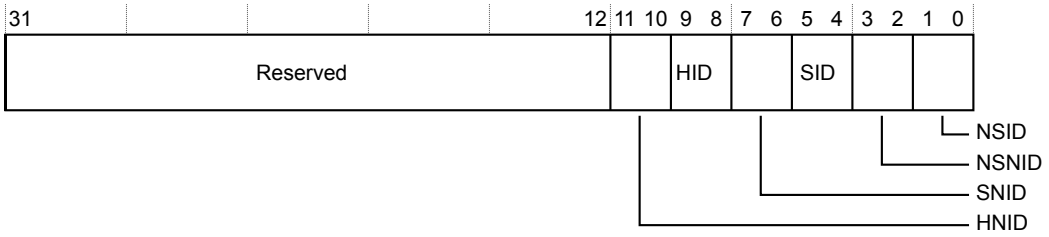

Figure 9-264 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-274 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	`0b00`	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-274 AUTHSTATUS register bit assignments (continued)

Bits	Reset value	Name	Function
[3:2]	0b00	NSNID	Non-secure non-invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.
[1:0]	0b00	NSID	Non-secure invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

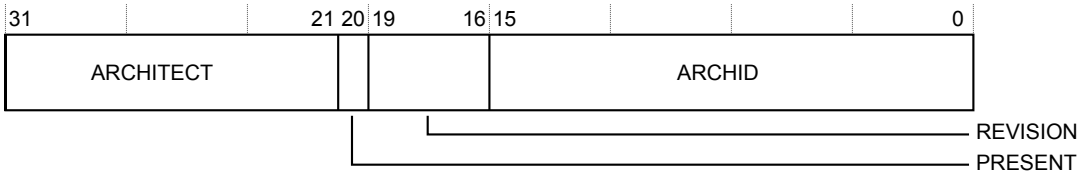

Figure 9-265 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-275 DEVARCH register bit assignments

Bits	Reset value	Name	Function
[31:21]	`0b00000000000`	ARCHITECT	Returns 0.
[20]	`0b0`	PRESENT	Returns 0, indicating that the DEVARCH register is not present.
[19:16]	`0b0000`	REVISION	Returns 0
[15:0]	`0x0`	ARCHID	Returns 0.

Device Configuration Register 2, DEVID2

Contains an IMPLEMENTATION DEFINED value.

The DEVID2 register characteristics are:

Attributes

Offset	`0x0fc0`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

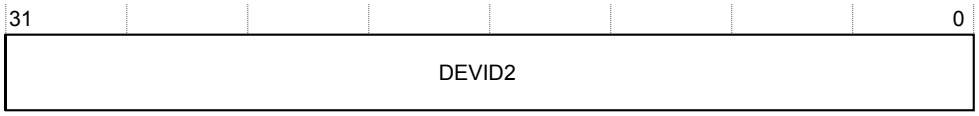

Figure 9-266 DEVID2 register bit assignments

The following table shows the bit assignments.

Table 9-276 DEVID2 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVID2	This field is RAZ.

Device Configuration Register 1, DEVID1

Contains an IMPLEMENTATION DEFINED value.

The DEVID1 register characteristics are:

Attributes

Offset	`0x0fc4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

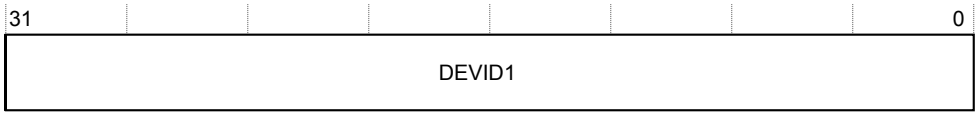

Figure 9-267 DEVID1 register bit assignments

The following table shows the bit assignments.

Table 9-277 DEVID1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	DEVID1	This field is RAZ.

Device Configuration Register, DEVID

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

The DEVID register characteristics are:

Attributes

Offset	`0x0fc8`
Type	Read-only
Reset	`0x00000032`
Width	32

The following figure shows the bit assignments.

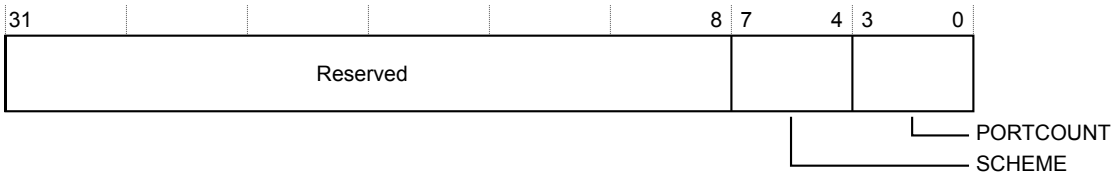

Figure 9-268 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-278 DEVID register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0011`	SCHEME	Indicates priority scheme implemented.
[3:0]	`0b0010`	PORTCOUNT	Indicates the number of master ports implemented.

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 `0x00000022`

Width 32

The following figure shows the bit assignments.

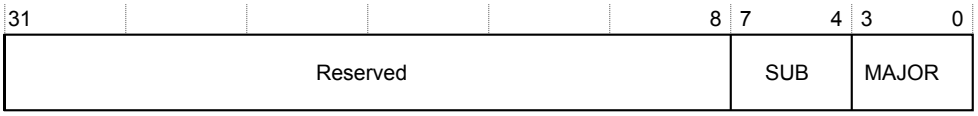

Figure 9-269 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-279 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0010`	SUB	Minor classification. Returns `0x2`, indicates this component is a Filter.
[3:0]	`0b0010`	MAJOR	Major classification. Returns `0x2`, indicating this component is a Trace Link.

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.

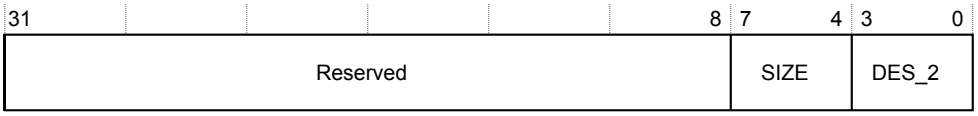

Figure 9-270 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-280 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

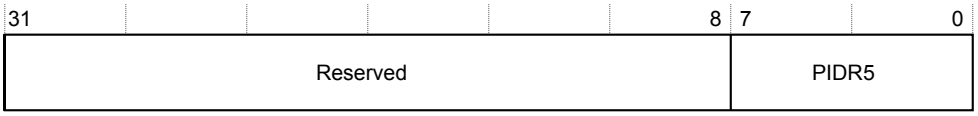

Figure 9-271 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-281 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

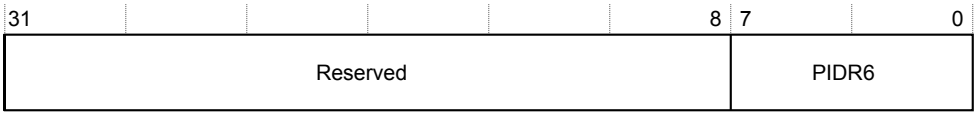

Figure 9-272 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-282 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

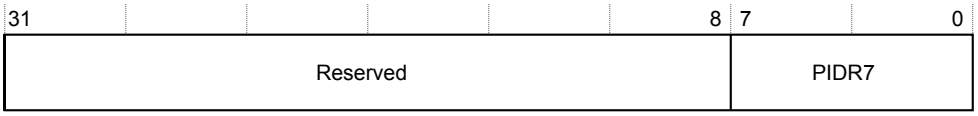

Figure 9-273 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-283 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000ec`

Width 32

The following figure shows the bit assignments.

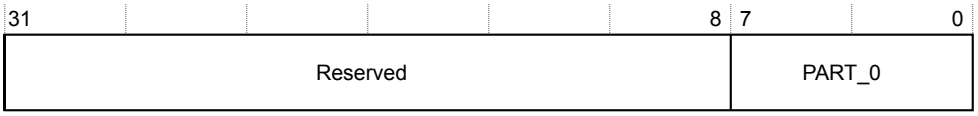

Figure 9-274 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-284 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11101100`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

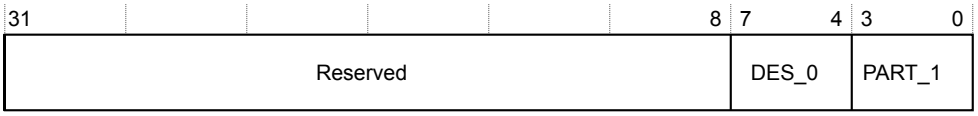

Figure 9-275 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-285 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x0000000b`
Width 32

The following figure shows the bit assignments.

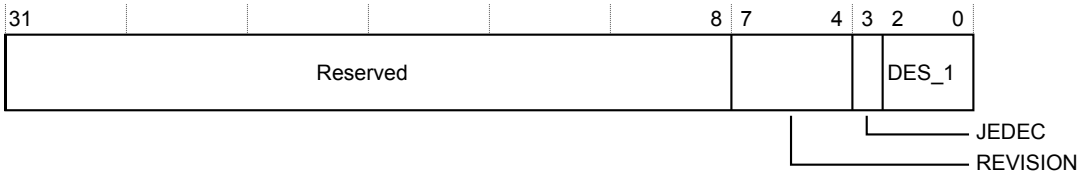

Figure 9-276 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-286 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

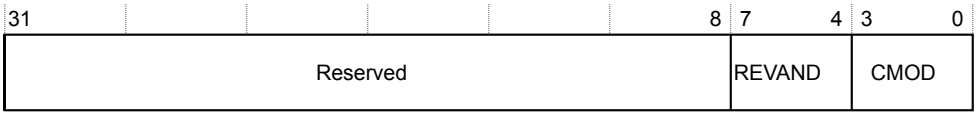

Figure 9-277 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-287 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

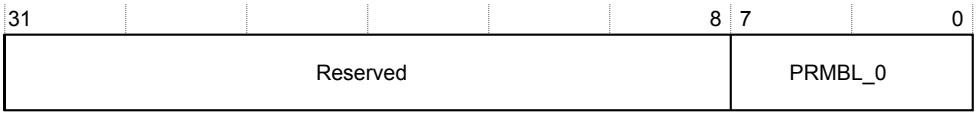

Figure 9-278 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-288 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

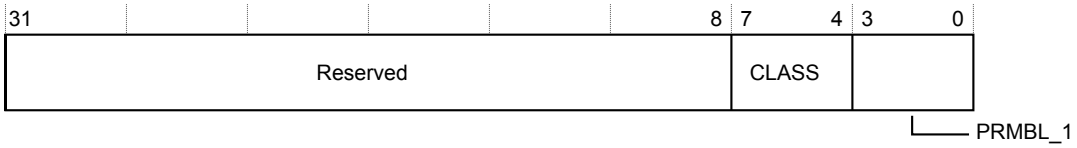

Figure 9-279 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-289 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

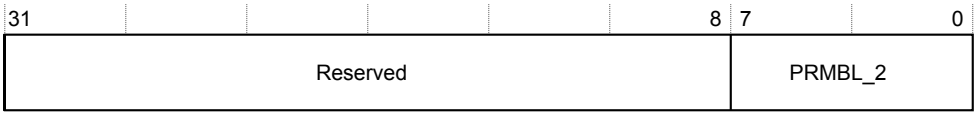

Figure 9-280 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-290 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

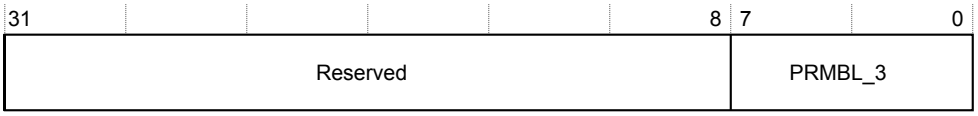

Figure 9-281 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-291 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.12 css600_tmc_etb introduction

This section describes the functions and programmers model of the `css600_tmc_etb`.

This section contains the following subsections:

- *9.12.1 Register summary* on page 9-424.
- *9.12.2 Register descriptions* on page 9-425.

9.12.1 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.

Table 9-292 css600_tmc_etb - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0004	RSZ	RW	0x------00	32	*RAM Size register, RSZ* on page 9-426
0x000C	STS	RW	0x000000--	32	*Status register, STS* on page 9-427
0x0010	RRD	RO	0x--------	32	*RAM Read Data register, RRD* on page 9-428
0x0014	RRP	RW	0x--------	32	*RAM Read Pointer register, RRP* on page 9-429
0x0018	RWP	RW	0x--------	32	*RAM Write Pointer register, RWP* on page 9-430
0x001C	TRG	RW	0x--------	32	*Trigger Counter register, TRG* on page 9-431
0x0020	CTL	RW	0x00000000	32	*Control Register, CTL* on page 9-432
0x0024	RWD	WO	0x00000000	32	*RAM Write Data register, RWD* on page 9-433
0x0028	MODE	RW	0x0000000-	32	*Mode register, MODE* on page 9-434
0x002C	LBUFLEVEL	RO	0x--------	32	*Latched Buffer Fill Level, LBUFLEVEL* on page 9-435
0x0030	CBUFLEVEL	RO	0x--------	32	*Current Buffer Fill Level, CBUFLEVEL* on page 9-436
0x0034	BUFWM	RW	0x--------	32	*Buffer Level Water Mark, BUFWM* on page 9-437
0x0300	FFSR	RO	0x0000000-	32	*Formatter and Flush Status Register, FFSR* on page 9-438
0x0304	FFCR	RW	0x00000000	32	*Formatter and Flush Control Register, FFCR* on page 9-439
0x0308	PSCR	RW	0x0000000a	32	*Periodic Synchronization Counter Register, PSCR* on page 9-442
0x0EE0	ITEVTINTR	WO	0x00000000	32	*Integration Test Event and Interrupt Control Register, ITEVTINTR* on page 9-443
0x0EE8	ITTRFLIN	RO	0x00000000	32	*Integration Test Trigger In and Flush In register, ITTRFLIN* on page 9-444
0x0EEC	ITATBDATA0	RO	0x00000000	32	*Integration Test ATB Data 0 Register, ITATBDATA0* on page 9-445
0x0EF0	ITATBCTR2	WO	0x00000000	32	*Integration Test ATB Control 2 Register, ITATBCTR2* on page 9-447
0x0EF4	ITATBCTR1	RO	0x00000000	32	*Integration Test ATB Control 1 Register, ITATBCTR1* on page 9-448
0x0EF8	ITATBCTR0	RO	0x00000000	32	*Integration Test ATB Control 0 Register, ITATBCTR0* on page 9-449
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-450

Table 9-292 css600_tmc_etb - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FA0	CLAIMSET	RW	0x0000000f	32	*Claim Tag Set Register, CLAIMSET* on page 9-451
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-452
0x0FB8	AUTHSTATUS	RO	0x00000000	32	*Authentication Status Register, AUTHSTATUS* on page 9-453
0x0FC8	DEVID	RO	0x00000-00	32	*Device Configuration Register, DEVID* on page 9-455
0x0FC4	DEVID1	RO	0x00000001	32	*Device Configuration Register, DEVID1* on page 9-456
0x0FCC	DEVTYPE	RO	0x00000021	32	*Device Type Identifier Register, DEVTYPE* on page 9-457
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-458
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-459
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-460
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-461
0x0FE0	PIDR0	RO	0x000000e9	32	*Peripheral Identification Register 0, PIDR0* on page 9-462
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-463
0x0FE8	PIDR2	RO	0x00000001	32	*Peripheral Identification Register 2, PIDR2* on page 9-464
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-465
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-466
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-467
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-468
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-469

9.12.2 Register descriptions

This section describes the `css600_tmc_etb` registers.

9.12.1 Register summary on page 9-424 provides cross references to individual registers.

RAM Size register, RSZ

Defines the size of trace memory in units of 32-bit words.

The RSZ register characteristics are:

Attributes

Offset `0x0004`

Type Read-write

Reset `0x------00`

Width 32

The following figure shows the bit assignments.

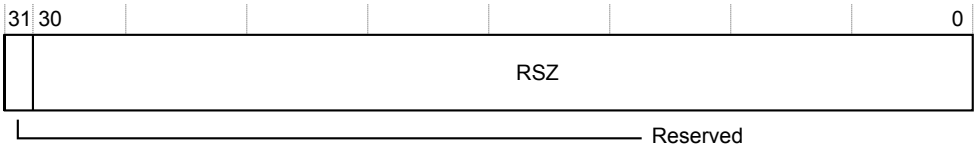

Figure 9-282 RSZ register bit assignments

The following table shows the bit assignments.

Table 9-293 RSZ register bit assignments

Bits	Reset value	Name	Function
[30:0]	IMPLEMENTATION_DEFINED	RSZ	RAM size. Indicates the size of the RAM in 32-bit words. Returns the programmed value on reading. For example: Returns `0x00000100` if trace memory size is 1KB, `0x40000000` if trace memory size is 4GB. This field has the same value as the `MEM_SIZE`parameter.

Status register, STS

Indicates the status of the Trace Memory Controller. After a reset, software must ignore all the fields of this register except STS.TMCReady. The other fields have meaning only when the TMC has left the Disabled state. Writes to all RO fields of this register are ignored.

The STS register characteristics are:

Attributes

Offset	`0x000c`
Type	Read-write
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

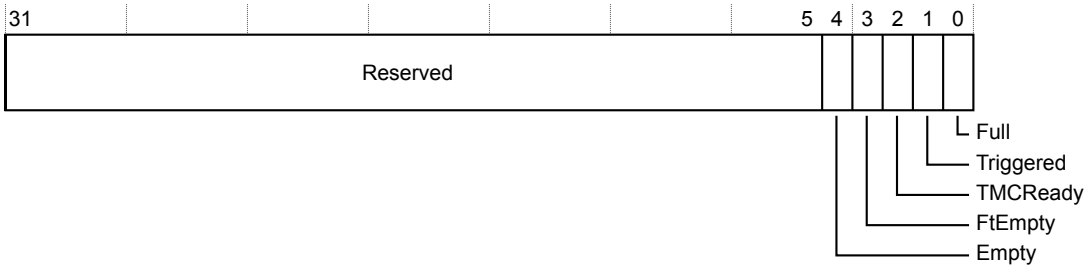

Figure 9-283 STS register bit assignments

The following table shows the bit assignments.

Table 9-294 STS register bit assignments

Bits	Reset value	Name	Function
[4]	UNKNOWN	Empty	Trace buffer empty. If set, this bit indicates that the trace memory does not contain any valid trace data. However, this does not mean that the pipeline stages within the TMC are empty. To determine whether the internal pipeline stages are empty, the software must read the STS.TMCReady bit.
[3]	UNKNOWN	FtEmpty	Formatter pipeline empty. This bit is deprecated and is present in this register to support backwards compatibility with earlier versions of the ETB. It is set when trace capture has stopped, and all internal pipelines and buffers have drained. Unlike STS.TMCReady, it is not affected by buffer drains and AXI accesses. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.
[2]	`0b1`	TMCReady	TMC ready. This flag is set when trace capture has stopped and all internal pipelines and buffers have drained, the TMC is not draining as a result of FFCR.DrainBuffer bit being set (ETF only), and the AXI interface is not busy and the response for final AXI write has been received (ETR only) are all true. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.
[1]	UNKNOWN	Triggered	TMC triggered. This bit is set when trace capture is in progress and the TMC has detected a trigger event. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. A trigger event is when the TMC has written a set number of data words, as programmed in the TRG register, into the trace memory after a rising edge of **trigin** input, or a trigger packet (**atid_s** = `0x7D`) is received in the input trace.
[0]	UNKNOWN	Full	Trace memory full. This bit helps in determining the amount of valid data present in the trace memory. It is not affected by the reprogramming of pointer registers in Disabled state. It is cleared when the TMC leaves Disabled state.

RAM Read Data register, RRD

Reading this register allows data to be read from the trace memory. When the memory width is greater than the APB width (32-bit), multiple reads to the same location must be performed to get the complete memory dataword. The RRP register is incremented only after the last of the required number of read accesses to fetch one memory dataword completes. When the trace memory is empty, this register returns `0xFFFFFFFF`. This value is chosen because it cannot be generated as part of trace data when the formatter is enabled.

The RRD register characteristics are:

Attributes

Offset `0x0010`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

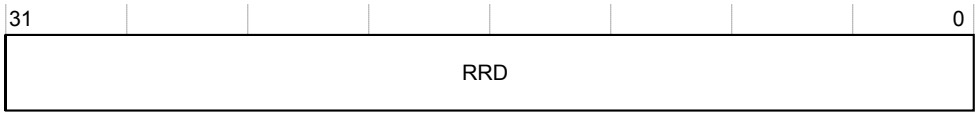

Figure 9-284 RRD register bit assignments

The following table shows the bit assignments.

Table 9-295 RRD register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RRD	Returns the data read from trace memory.

RAM Read Pointer register, RRP

The RAM Read Pointer Register contains the value of the read pointer that is used to read entries from trace memory over the APB interface. Software must program it before enabling trace capture.

The RRP register characteristics are:

Attributes

Offset `0x0014`
Type Read-write
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

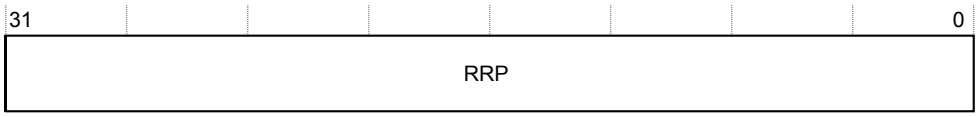

Figure 9-285 RRP register bit assignments

The following table shows the bit assignments.

Table 9-296 RRP register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RRP	The RRP width depends on the size of trace memory and is given by $\log_2$(`MEM_SIZE`x 4). The remaining MSBs of the 32-bit register are of type RAZ/WI.

RAM Write Pointer register, RWP

RAM Write Pointer Register sets the write pointer that is used to write entries into the trace memory. Software must program it before enabling trace capture.

The RWP register characteristics are:

Attributes

Offset `0x0018`

Type Read-write

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

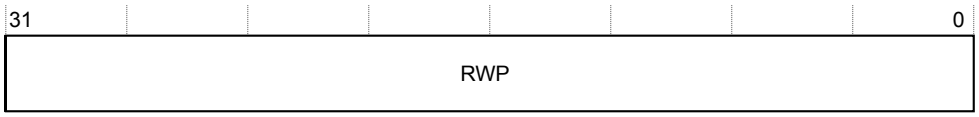

Figure 9-286 RWP register bit assignments

The following table shows the bit assignments.

Table 9-297 RWP register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RWP	The RWP width depends on the size of trace memory and is given by $\log_2$(`MEM_SIZE`x 4). The remaining MSBs of the 32-bit register are of type RAZ/WI.

Trigger Counter register, TRG

In Circular Buffer mode, the Trigger Counter register specifies the number of 32-bit words to capture in the trace memory, after detection of either a rising edge on the **trigin** input or a trigger packet in the incoming trace stream, that is, where **atid_s** = `0x7D`. The value programmed must be aligned to the frame length of 128 bits.

The TRG register characteristics are:

Attributes

Offset	`0x001c`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

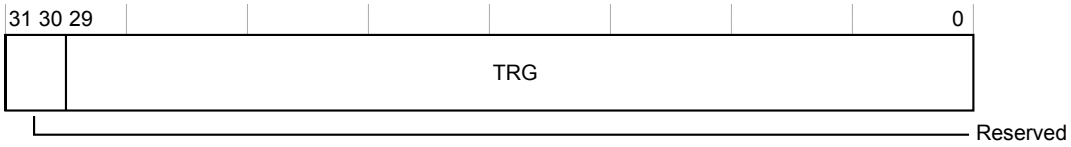

Figure 9-287 TRG register bit assignments

The following table shows the bit assignments.

Table 9-298 TRG register bit assignments

Bits	Reset value	Name	Function
[29:0]	UNKNOWN	TRG	Trigger count. This count represents the number of 32-bit words of trace that are captured between a trigger packet and a trigger event. Software must program this register before setting CTL.TraceCaptEn bit if operating in CB mode.

Control Register, CTL

This register controls trace stream capture. Setting the CTL.TraceCaptEn bit to 1 enables the TMC to capture the trace data. When trace capture is enabled, the formatter behavior is controlled by the FFCR register.

The CTL register characteristics are:

Attributes

Offset `0x0020`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

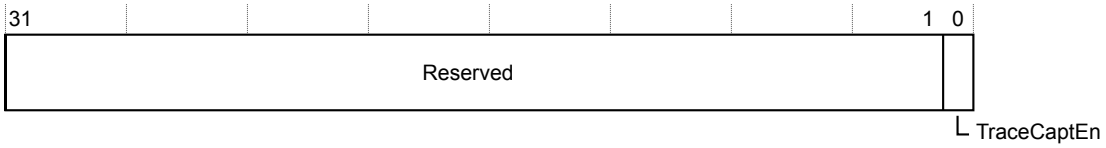

Figure 9-288 CTL register bit assignments

The following table shows the bit assignments.

Table 9-299 CTL register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	TraceCaptEn	Trace capture enable. **0** Disable trace capture. **1** Enable trace capture.

RAM Write Data register, RWD

The RAM Write Data register enables testing of trace memory connectivity to the TMC. When in Disabled state, a write to this register stores data at the location pointed to by the RWP/RWPHI registers. When not in Disabled state, writes are ignored. When the memory data width, as indicated by the DEVID.MEM_WIDTH register field, is greater than 32 bits, multiple writes to this register must be performed together to write a full memory width of data. For example, if the memory width is 128 bits, then writes to this register must be performed 4 at a time. When a full memory width of data has been written to this register, the data is written to the trace memory and the RWP is incremented to the next memory word.

The RWD register characteristics are:

Attributes

Offset `0x0024`
Type Write-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

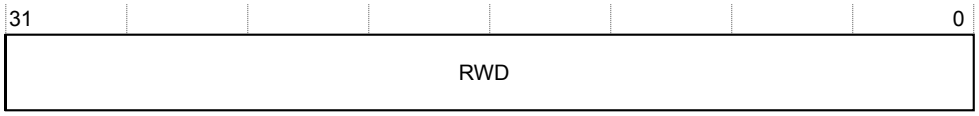

Figure 9-289 RWD register bit assignments

The following table shows the bit assignments.

Table 9-300 RWD register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	RWD	Data written to this register is placed in the trace memory.

Mode register, MODE

This register controls the TMC operating mode. The operating mode can only be changed when the TMC is in Disabled state, that is, when CTL.TraceCaptEn = 0. Attempting to write to this register in any other state results in UNPREDICTABLE behavior. The operating mode is ignored when in Disabled state.

The MODE register characteristics are:

Attributes

Offset `0x0028`
Type Read-write
Reset `0x0000000-`
Width 32

The following figure shows the bit assignments.

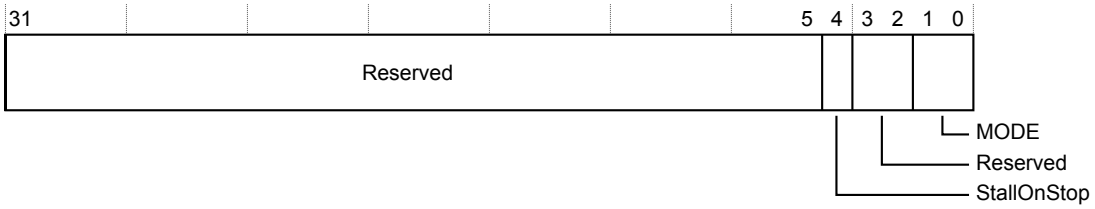

Figure 9-290 MODE register bit assignments

The following table shows the bit assignments.

Table 9-301 MODE register bit assignments

Bits	Reset value	Name	Function
[4]	`0b0`	StallOnStop	Stall On Stop. If this bit is set and the formatter stops as a result of a stop event, the output **atready_s** is de-asserted to stall the ATB interface and avoid loss of trace. If this bit is clear and the formatter stops as a result of a stop event, signal **atready_s** remains asserted but the TMC discards further incoming trace.
[1:0]	UNKNOWN	MODE	Selects the operating mode. If a reserved MODE value is programmed and trace capture is enabled, the TMC starts to operate in SWF1 mode. However, reading the MODE.MODE field returns the programmed value. `0x0` CB, Circular Buffer mode. `0x1` SWF1, Software Read FIFO mode 1. `0x2` Reserved. `0x3` Reserved.

Latched Buffer Fill Level, LBUFLEVEL

Reading this register returns the maximum fill level of the trace memory in 32-bit words since this register was last read. Reading this register also results in its contents being updated to the current fill level. When entering Disabled state, it retains its last value. While in Disabled state, reads from this register do not affect its value. When exiting Disabled state, the LBUFLEVEL register is updated to the current fill level.

The LBUFLEVEL register characteristics are:

Attributes

Offset `0x002c`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

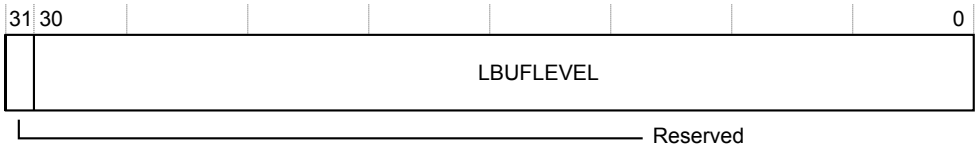

Figure 9-291 LBUFLEVEL register bit assignments

The following table shows the bit assignments.

Table 9-302 LBUFLEVEL register bit assignments

Bits	Reset value	Name	Function
[30:0]	UNKNOWN	LBUFLEVEL	Latched Buffer Fill Level. Indicates the maximum fill level of the trace memory in 32-bit words since this register was last read. The width of the register is 1 + $\log_2$(`MEM_SIZE`).

Current Buffer Fill Level, CBUFLEVEL

The CBUFLEVEL register indicates the current fill level of the trace memory in units of 32-bit words. When the TMC leaves Disabled state, this register dynamically indicates the current fill level of trace memory. It retains its value on entering Disabled state. It is not affected by the reprogramming of pointer registers in Disabled state. However, RRD reads when using the Circular Buffer in Disabled state are a special case. In this case the register indicates the updated fill level on RRD reads.

The CBUFLEVEL register characteristics are:

Attributes

Offset `0x0030`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

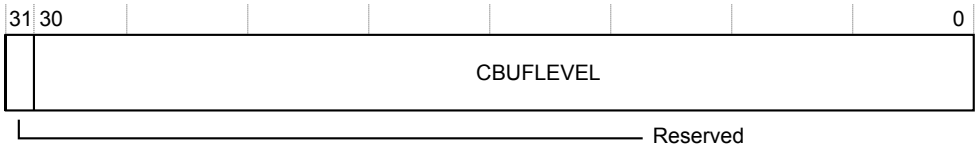

Figure 9-292 CBUFLEVEL register bit assignments

The following table shows the bit assignments.

Table 9-303 CBUFLEVEL register bit assignments

Bits	Reset value	Name	Function
[30:0]	UNKNOWN	CBUFLEVEL	Current Buffer Fill Level. Indicates the current fill level of the trace memory in 32-bit words. The width of the register is $1 + \log_2$(`MEM_SIZE`).

Buffer Level Water Mark, BUFWM

The value that is programmed into this register indicates the desired threshold vacancy level in 32-bit words in the trace memory. When the available space in the FIFO is less than or equal to this value, that is, fill level >= (MEM_SIZE- BUFWM), the **full** output is asserted and the STS.Full bit is set. This register is used only in the FIFO modes, that is, SWF1, SWF2, and HWF modes. In CB mode, the same functionality is obtained by programming the RWP to the desired vacancy trigger level, so that when the pointer wraps around, the **full** output gets asserted indicating that the vacancy level has fallen below the desired level. Reading this register returns the programmed value. The maximum value that can be written into this register is MEM_SIZE- 1, in which case the **full** output is asserted after the first 32-bit word is written to trace memory. Writing to this register other than when in Disabled state results in UNPREDICTABLE behavior. Any software using it must program it with an initial value before setting the CTL.TraceCaptEn bit to 1.

The BUFWM register characteristics are:

Attributes

Offset	0x0034
Type	Read-write
Reset	0x--------
Width	32

The following figure shows the bit assignments.

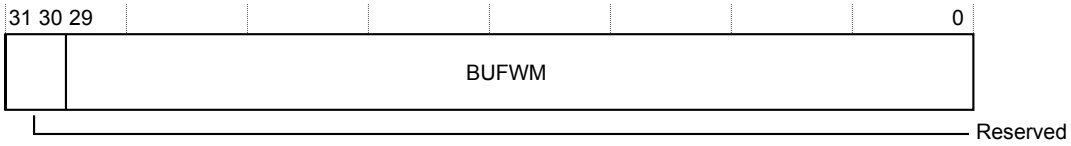

Figure 9-293 BUFWM register bit assignments

The following table shows the bit assignments.

Table 9-304 BUFWM register bit assignments

Bits	Reset value	Name	Function
[29:0]	UNKNOWN	BUFWM	Buffer Level Watermark. Indicates the desired threshold vacancy level in 32-bit words in the trace memory. The width of the register is $\log_2$(MEM_SIZE).

Formatter and Flush Status Register, FFSR

This register indicates the status of the Formatter, and the status of Flush request.

The FFSR register characteristics are:

Attributes

Offset	`0x0300`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

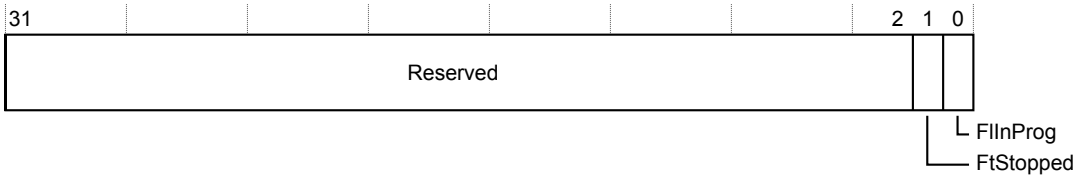

Figure 9-294 FFSR register bit assignments

The following table shows the bit assignments.

Table 9-305 FFSR register bit assignments

Bits	Reset value	Name	Function
[1]	UNKNOWN	FtStopped	Formatter Stopped. This bit behaves the same way as STS.FtEmpty. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. The FFCR.FtStopped bit is deprecated and is present in this register to support backwards-compatibility with earlier versions of the ETB. **0** Trace capture has not yet completed. **1** Trace capture has completed and all captured trace data has been written to the trace memory.
[0]	UNKNOWN	FlInProg	Flush In Progress. This bit indicates whether the TMC is currently processing a flush request. Flush initiation is controlled by the flush control bits in the FFCR register. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. When in Disabled state, this bit is not updated. **0** No flush activity in progress. **1** Flush in progress on the ATB slave interface or the TMC internal pipeline.

Formatter and Flush Control Register, FFCR

This register allows user control of the stop, trigger, and flush events.

The FFCR register characteristics are:

Attributes

Offset `0x0304`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

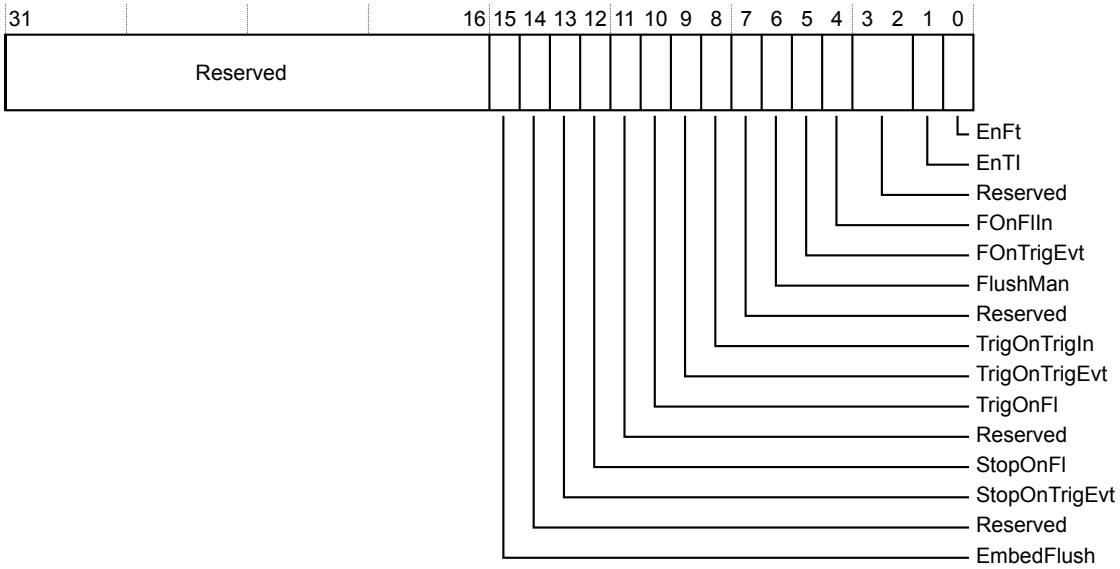

Figure 9-295 FFCR register bit assignments

The following table shows the bit assignments.

Table 9-306 FFCR register bit assignments

Bits	Reset value	Name	Function
[15]	`0b0`	EmbedFlush	Embed Flush ID (flush completion packet). Enables insertion of Flush ID `0x7B` with a single byte of data payload = `0x00` in the output trace, immediately after the last flush data byte, when a flush completes on the ATB slave interface. This bit is effective only in Normal formatting modes. In Bypass mode, the Flush ID insertion remains disabled and this bit is ignored. **0** Disable Flush ID insertion. **1** Enable Flush ID insertion.
[13]	`0b0`	StopOnTrigEvt	Stop On Trigger Event. If this bit is set, the formatter is stopped when a Trigger Event has been observed. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.

Table 9-306 FFCR register bit assignments (continued)

Bits	Reset value	Name	Function
[12]	0b0	StopOnFl	Stop On Flush. If this bit is set, the formatter is stopped on completion of a flush operation. The initiation of a flush operation is controlled by programming the register bits FFCR.FlushMan, FFCR.FOnTrigEvt, and FFCR.FOnFlIn. When a flush-initiation condition occurs, **afvalid_s** is asserted, and when the flush completion is received, that is, **afready_s**=1, trace capture is stopped. Any remaining data in the formatter is appended with a post-amble and written to trace memory. The flush operation is then complete. When the TMC is configured as an ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a formatter stop regardless of the value that is programmed in this bit.
[10]	0b0	TrigOnFl	Indicate on trace stream the completion of flush. If this bit is set, a trigger is indicated on the trace stream when **afready_s** is received for a flush in progress. If this bit is clear, no triggers are embedded in the trace stream on flush completion. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. When the TMC is configured as ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a trigger indication on the trace stream regardless of the value that is programmed in this bit.
[9]	0b0	TrigOnTrigEvt	Indicate on trace stream the occurrence of a Trigger Event. If this bit is set, a trigger is indicated on the output trace stream when a Trigger Event occurs. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[8]	0b0	TrigOnTrigIn	Indicate on trace stream the occurrence of a rising edge on **trigin**. If this bit is set, a trigger is indicated on the trace stream when a rising edge is detected on the **trigin** input. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit.
[6]	0b0	FlushMan	Manually generate a flush of the system. Writing 1 to this bit causes a flush to be generated. This bit is cleared automatically when, in formatter bypass mode, **afready_s** was sampled high, or, in normal formatting mode, **afready_s** was sampled high and all flush data was output to the trace memory. If CTL.TraceCaptEn=0, writes to this bit are ignored.
[5]	0b0	FOnTrigEvt	Flush on Trigger Event. Setting this bit generates a flush when a Trigger Event occurs. If FFCR.StopOnTrigEvt is set, this bit is ignored. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[4]	0b0	FOnFlIn	Setting this bit enables the detection of transitions on the **flushin** input by the TMC. If this bit is set and the formatter has not already stopped, a rising edge on **flushin** initiates a flush request.

Table 9-306 FFCR register bit assignments (continued)

Bits	Reset value	Name	Function
[1]	0b0	EnTI	Enable Trigger Insertion. Setting this bit enables the insertion of triggers in the formatted trace stream. A trigger is indicated by inserting one byte of data 0x00 with **atid_s**=0x7D in the trace stream. Trigger indication on the trace stream is also controlled by the register bits FFCR.TrigOnFl, FFCR.TrigOnTrigEvt, and FFCR.TrigOnTrigIn. This bit can only be changed when the TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled.
[0]	0b0	EnFt	Enable Formatter. If this bit is set, formatting is enabled. This bit takes effect when not in Disabled state, and is ignored in Disabled state. If this bit is clear, formatting is disabled. In this case, the incoming trace data is assumed to be from a single trace source. If multiple trace IDs are received by the TMC when trace capture is enabled and the formatter is disabled, it results in interleaving of trace data. Disabling of formatting is deprecated, and is supported in the TMC for backwards-compatibility with earlier versions of the ETB. Therefore, disabling of formatting is supported only in CB mode. Features in the TMC such as the FIFO modes and the FFCR.DrainBuffer bit that are not part of the earlier versions of the ETB do not support disabling of formatting. This bit can only be changed when TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled. If the TMC is enabled in a mode other than Circular Buffer mode with formatting disabled, it results in formatting being enabled.

Periodic Synchronization Counter Register, PSCR

This register determines the reload value of the Periodic Synchronization Counter. This counter enables the frequency of sync packets to be optimized to the trace capture buffer size. The default behavior of the counter is to generate periodic synchronization requests, **syncreq_s**, on the ATB slave interface.

The PSCR register characteristics are:

Attributes

Offset `0x0308`
Type Read-write
Reset `0x0000000a`
Width 32

The following figure shows the bit assignments.

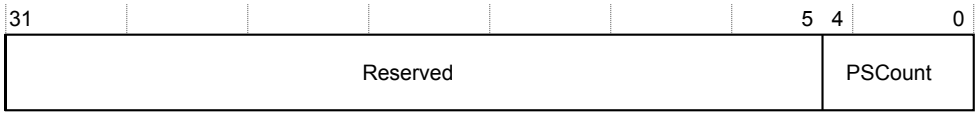

Figure 9-296 PSCR register bit assignments

The following table shows the bit assignments.

Table 9-307 PSCR register bit assignments

Bits	Reset value	Name	Function
[4:0]	`0b01010`	PSCount	Periodic Synchronization Count. Determines the reload value of the Synchronization Counter. The reload value takes effect the next time the counter reaches zero. When trace capture is enabled, the Synchronization Counter counts the number of bytes of trace data that is stored into the trace memory, regardless of whether the trace data has been formatted by the TMC or not, since the occurrence of the last sync request on the ATB slave interface. When the counter reaches 0, a sync request is sent on the ATB slave interface. Reads from this register return the reload value that is programmed in this register. This field resets to `0x0A`, that is, the default sync period is 2^10 bytes. If a reserved value is programmed in this register field, the value `0x1B` is used instead, and subsequent reads from this register also return `0x1B`. The following constraints apply to the values written to the PSCount field: `0x0` - synchronization is disabled, `0x1-0x6` - reserved, `0x7-0x1B` - synchronization period is 2^PSCount bytes. The smallest value `0x7` gives a sync period of 128 bytes. The maximum allowed value `0x1B` gives a sync period of 2^27 bytes, `0x1C-0x1F` - reserved.

Integration Test Event and Interrupt Control Register, ITEVTINTR

This register controls the values of event and interrupt outputs in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITEVTINTR register characteristics are:

Attributes

Offset	`0x0ee0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

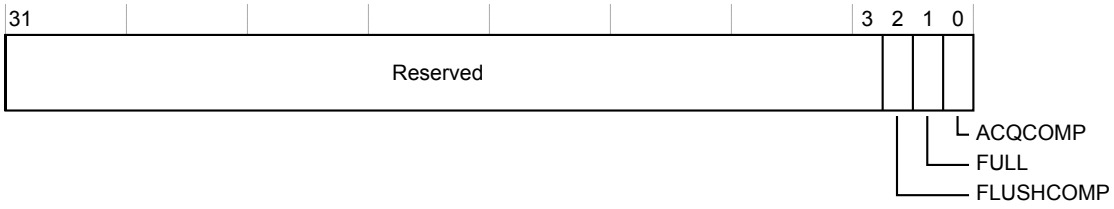

Figure 9-297 ITEVTINTR register bit assignments

The following table shows the bit assignments.

Table 9-308 ITEVTINTR register bit assignments

Bits	Reset value	Name	Function
[2]	`0b0`	FLUSHCOMP	Controls the value of **flushcomp** output in integration mode.
[1]	`0b0`	FULL	Controls the value of **full** output in integration mode.
[0]	`0b0`	ACQCOMP	Controls the value of **acqcomp** output in integration mode.

Integration Test Trigger In and Flush In register, ITTRFLIN

This register captures the values of the **flushin** and **trigin** inputs in integration mode. In functional mode, this register behaves as RAZ/WI.

The ITTRFLIN register characteristics are:

Attributes

Offset	`0x0ee8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

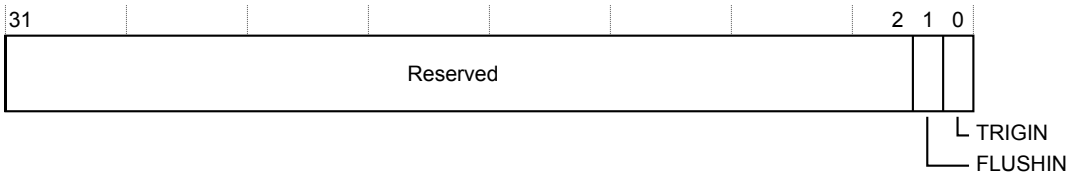

Figure 9-298 ITTRFLIN register bit assignments

The following table shows the bit assignments.

Table 9-309 ITTRFLIN register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	FLUSHIN	Integration status of **flushin** input. In integration mode, this bit latches to 1 on a rising edge of the **flushin** input. It is cleared when the register is read or when integration mode is disabled.
[0]	`0b0`	TRIGIN	Integration status of **trigin** input. In integration mode, this bit latches to 1 on a rising edge of the **trigin** input. It is cleared when the register is read or when integration mode is disabled.

Integration Test ATB Data 0 Register, ITATBDATA0

This register captures the value of **atdata_s** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding **atdata_s** bits. The width of this register is given by: 1+(ATB DATA WIDTH)/8.

The ITATBDATA0 register characteristics are:

Attributes

Offset `0x0eec`
Type Read-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

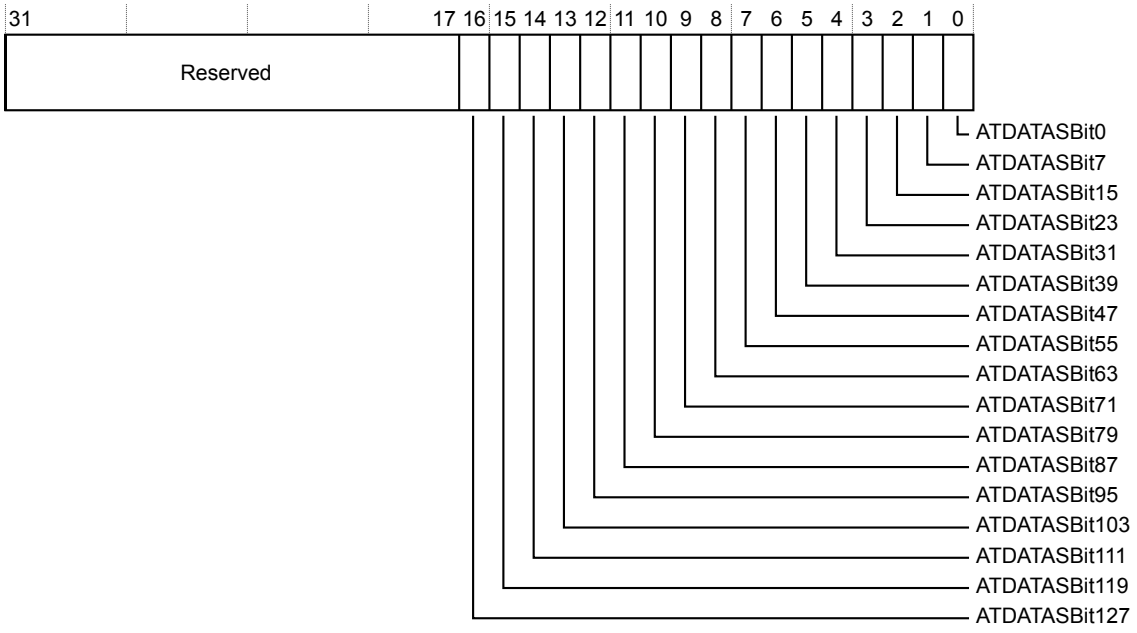

Figure 9-299 ITATBDATA0 register bit assignments

The following table shows the bit assignments.

Table 9-310 ITATBDATA0 register bit assignments

Bits	Reset value	Name	Function
[16]	`0b0`	ATDATASBit127	Returns the value of **atdata_s[127]** input in integration mode.
[15]	`0b0`	ATDATASBit119	Returns the value of **atdata_s[119]** input in integration mode.
[14]	`0b0`	ATDATASBit111	Returns the value of **atdata_s[111]** input in integration mode.
[13]	`0b0`	ATDATASBit103	Returns the value of **atdata_s[103]** input in integration mode.
[12]	`0b0`	ATDATASBit95	Returns the value of **atdata_s[95]** input in integration mode.
[11]	`0b0`	ATDATASBit87	Returns the value of **atdata_s[87]** input in integration mode.
[10]	`0b0`	ATDATASBit79	Returns the value of **atdata_s[79]** input in integration mode.
[9]	`0b0`	ATDATASBit71	Returns the value of **atdata_s[71]** input in integration mode.
[8]	`0b0`	ATDATASBit63	Returns the value of **atdata_s[63]** input in integration mode.

Table 9-310 ITATBDATA0 register bit assignments (continued)

Bits	Reset value	Name	Function
[7]	0b0	ATDATASBit55	Returns the value of **atdata_s[55]** input in integration mode.
[6]	0b0	ATDATASBit47	Returns the value of **atdata_s[47]** input in integration mode.
[5]	0b0	ATDATASBit39	Returns the value of **atdata_s[39]** input in integration mode.
[4]	0b0	ATDATASBit31	Returns the value of **atdata_s[31]** input in integration mode.
[3]	0b0	ATDATASBit23	Returns the value of **atdata_s[23]** input in integration mode.
[2]	0b0	ATDATASBit15	Returns the value of **atdata_s[15]** input in integration mode.
[1]	0b0	ATDATASBit7	Returns the value of **atdata_s[7]** input in integration mode.
[0]	0b0	ATDATASBit0	Returns the value of **atdata_s[0]** input in integration mode.

Integration Test ATB Control 2 Register, ITATBCTR2

This register enables control of ATB slave outputs **atready_s**, **afvalid_s**, and **syncreq_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITATBCTR2 register characteristics are:

Attributes

Offset	`0x0ef0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

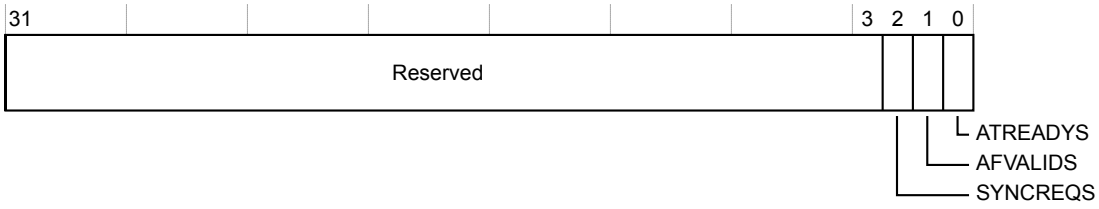

Figure 9-300 ITATBCTR2 register bit assignments

The following table shows the bit assignments.

Table 9-311 ITATBCTR2 register bit assignments

Bits	Reset value	Name	Function
[2]	`0b0`	SYNCREQS	Controls the value of **syncreq_s** output in integration mode.
[1]	`0b0`	AFVALIDS	Controls the value of **afvalid_s** output in integration mode.
[0]	`0b0`	ATREADYS	Controls the value of **atready_s** output in integration mode.

Integration Test ATB Control 1 Register, ITATBCTR1

This register captures the value of the **atid_s[6:0]** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of **atid_s** input.

The ITATBCTR1 register characteristics are:

Attributes

Offset	`0x0ef4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

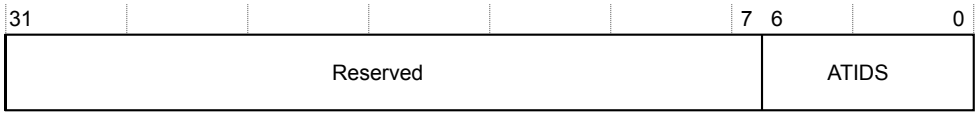

Figure 9-301 ITATBCTR1 register bit assignments

The following table shows the bit assignments.

Table 9-312 ITATBCTR1 register bit assignments

Bits	Reset value	Name	Function
[6:0]	`0b0000000`	ATIDS	Returns the value of **atid_s[6:0]** input in integration mode.

Integration Test ATB Control 0 Register, ITATBCTR0

This register captures the values of ATB slave inputs **atvalid_s**, **afready_s**, **atwakeup_s**, and **atbytes_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding input pins. The width of this register is given by: $8+\log_2(\text{ATB DATA WIDTH}/8)$.

The ITATBCTR0 register characteristics are:

Attributes

Offset	`0x0ef8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

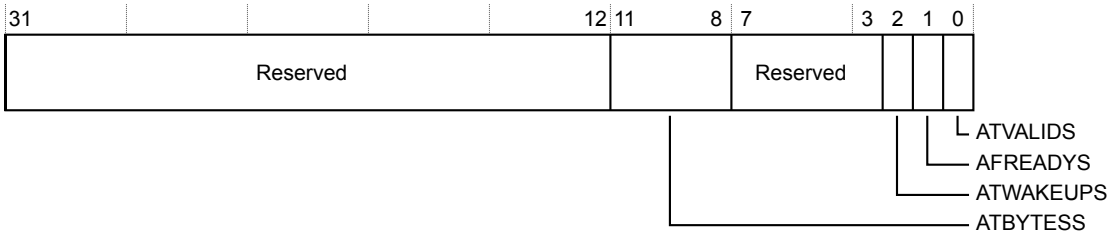

Figure 9-302 ITATBCTR0 register bit assignments

The following table shows the bit assignments.

Table 9-313 ITATBCTR0 register bit assignments

Bits	Reset value	Name	Function
[11:8]	`0b0000`	ATBYTESS	Returns the value of **atbytes_s** input in integration mode. $N=8+\log_2(\text{ATB DATA WIDTH}/8)$.
[2]	`0b0`	ATWAKEUPS	Returns the value of **atwakeup_s** input in integration mode.
[1]	`0b0`	AFREADYS	Returns the value of **afready_s** input in integration mode.
[0]	`0b0`	ATVALIDS	Returns the value of **atvalid_s** input in integration mode.

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.

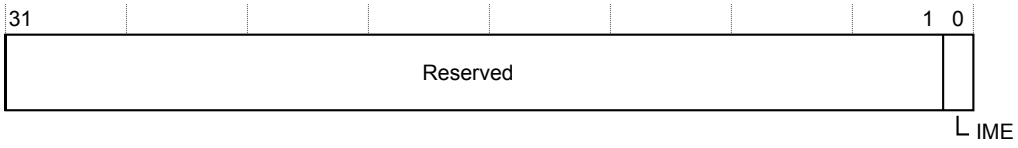

Figure 9-303 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-314 ITCTRL register bit assignments

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

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 `0x0000000f`
Width 32

The following figure shows the bit assignments.

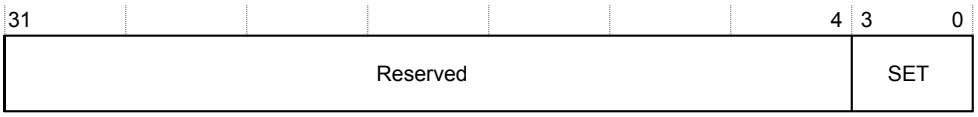

Figure 9-304 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-315 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	SET	A bit programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

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.

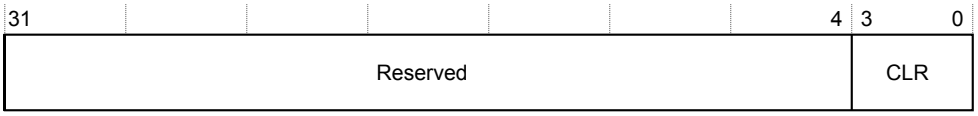

Figure 9-305 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-316 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	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.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

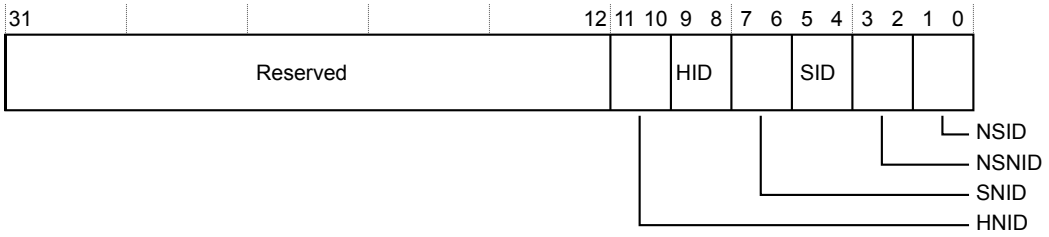

Figure 9-306 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-317 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	`0b00`	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-317 AUTHSTATUS register bit assignments (continued)

Bits	Reset value	Name	Function
[3:2]	0b00	NSNID	Non-secure non-invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.
[1:0]	0b00	NSID	Non-secure invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.

Device Configuration Register, DEVID

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

The DEVID register characteristics are:

Attributes

Offset `0x0fc8`
Type Read-only
Reset `0x00000-00`
Width 32

The following figure shows the bit assignments.

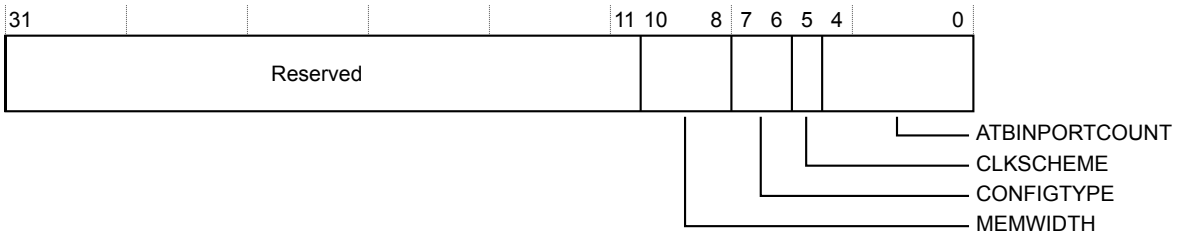

Figure 9-307 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-318 DEVID register bit assignments

Bits	Reset value	Name	Function
[10:8]	IMPLEMENTATION_DEFINED	MEMWIDTH	This value is twice `ATB_DATA_WIDTH`. `0x2` Memory interface databus is 32-bits wide. `0x3` Memory interface databus is 64-bits wide. `0x4` Memory interface databus is 128-bits wide. `0x5` Memory interface databus is 256-bits wide.
[7:6]	`0b00`	CONFIGTYPE	Returns `0x1`, indicating ETB configuration.
[5]	`0b0`	CLKSCHEME	RAM Clocking Scheme. This value indicates the TMC RAM clocking scheme used, that is, whether the TMC RAM operates synchronously or asynchronously to the TMC clock. Fixed to 0 indicating that TMC RAM clock is synchronous to the **clk** input.
[4:0]	`0b00000`	ATBINPORTCOUNT	Hidden Level of ATB input multiplexing. This value indicates the type/number of ATB multiplexing present on the input ATB. Fixed to `0x00` indicating that no multiplexing is present.

Device Configuration Register, DEVID1

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

The DEVID1 register characteristics are:

Attributes

Offset `0x0fc4`

Type Read-only

Reset `0x00000001`

Width 32

The following figure shows the bit assignments.

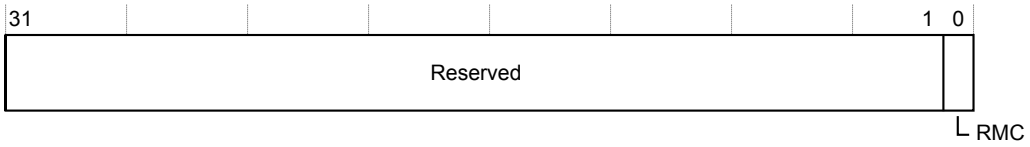

Figure 9-308 DEVID1 register bit assignments

The following table shows the bit assignments.

Table 9-319 DEVID1 register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x00000000`	-	Reserved.
[0]	1	RMC	Indicates register management mode 1 is implemented.

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 `0x00000021`
Width 32

The following figure shows the bit assignments.

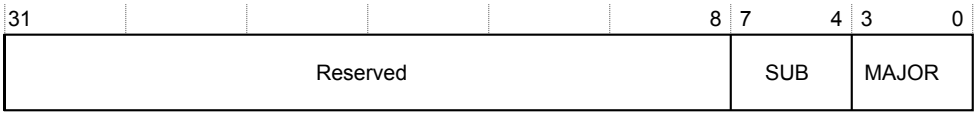

Figure 9-309 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-320 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0010`	SUB	Minor classification. Returns `0x2`, indicating this component is a Buffer.
[3:0]	`0b0001`	MAJOR	Major classification. Returns `0x1`, indicating this component is a Trace Sink.

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.

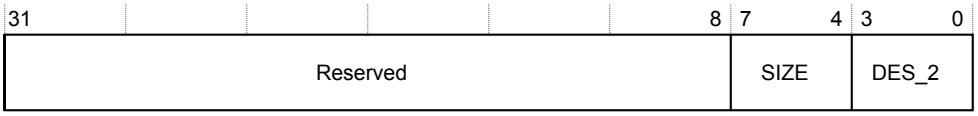

Figure 9-310 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-321 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

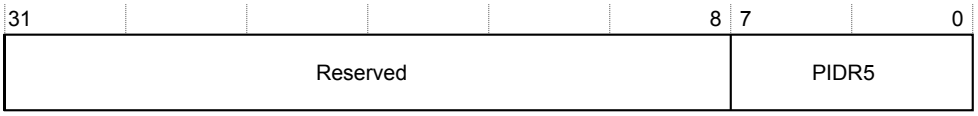

Figure 9-311 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-322 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

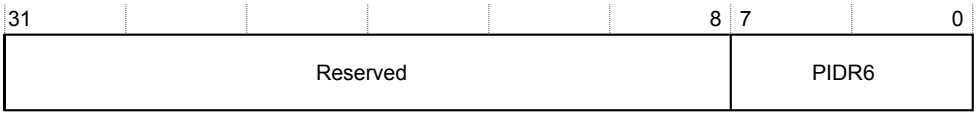

Figure 9-312 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-323 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

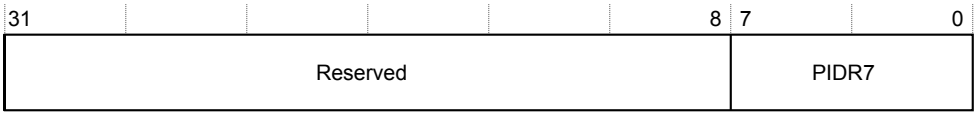

Figure 9-313 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-324 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000e9`

Width 32

The following figure shows the bit assignments.

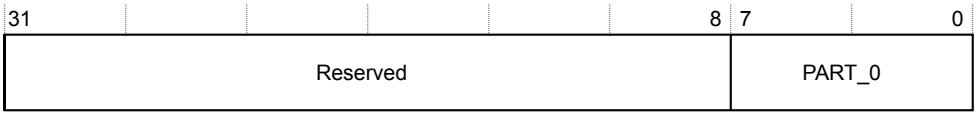

Figure 9-314 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-325 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11101001`	PART_0	Part number (lower 8 bits). Returns `0xe9`, indicating TMC ETB.

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.

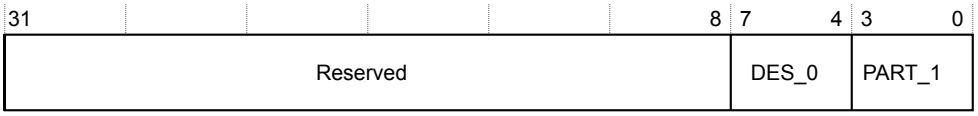

Figure 9-315 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-326 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x00000001`
Width 32

The following figure shows the bit assignments.

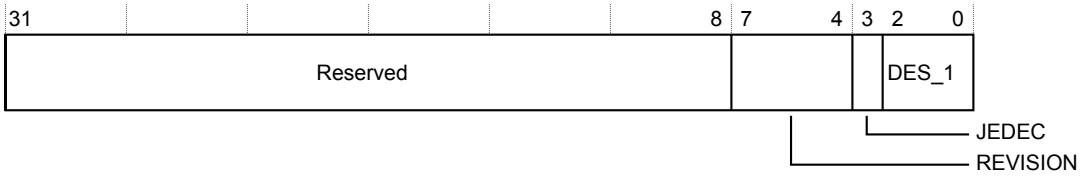

Figure 9-316 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-327 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0001`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

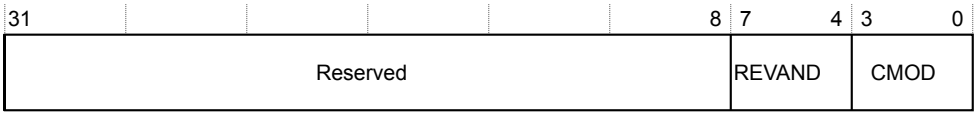

Figure 9-317 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-328 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

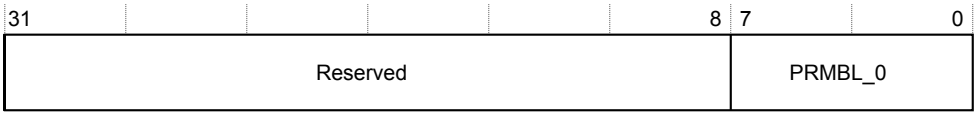

Figure 9-318 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-329 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

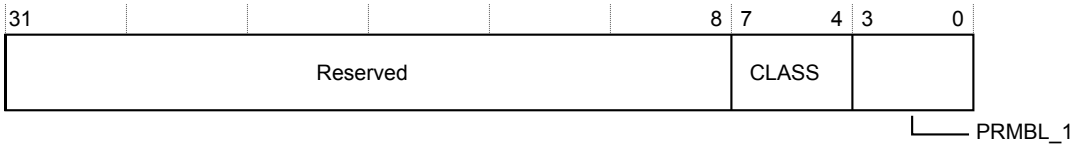

Figure 9-319 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-330 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

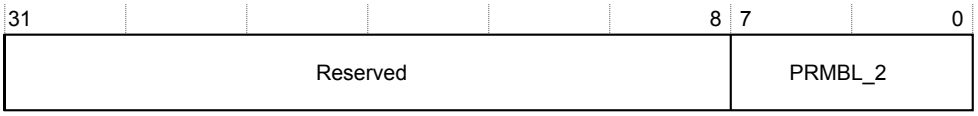

Figure 9-320 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-331 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

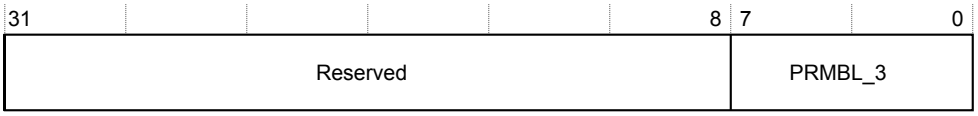

Figure 9-321 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-332 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.13 css600_tmc_etf introduction

This section describes the functions and programmers model of the `css600_tmc_etf`.

This section contains the following subsections:

- *9.13.1 Register summary* on page 9-470.
- *9.13.2 Register descriptions* on page 9-471.

9.13.1 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.

Table 9-333 css600_tmc_etf - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0004	RSZ	RW	0x------00	32	*RAM Size register, RSZ* on page 9-472
0x000C	STS	RW	0x000000--	32	*Status register, STS* on page 9-473
0x0010	RRD	RO	0x--------	32	*RAM Read Data register, RRD* on page 9-474
0x0014	RRP	RW	0x--------	32	*RAM Read Pointer register, RRP* on page 9-475
0x0018	RWP	RW	0x--------	32	*RAM Write Pointer register, RWP* on page 9-476
0x001C	TRG	RW	0x--------	32	*Trigger Counter register, TRG* on page 9-477
0x0020	CTL	RW	0x00000000	32	*Control Register, CTL* on page 9-478
0x0024	RWD	WO	0x00000000	32	*RAM Write Data register, RWD* on page 9-479
0x0028	MODE	RW	0x0000000-	32	*Mode register, MODE* on page 9-480
0x002C	LBUFLEVEL	RO	0x--------	32	*Latched Buffer Fill Level, LBUFLEVEL* on page 9-481
0x0030	CBUFLEVEL	RO	0x--------	32	*Current Buffer Fill Level, CBUFLEVEL* on page 9-482
0x0034	BUFWM	RW	0x--------	32	*Buffer Level Water Mark, BUFWM* on page 9-483
0x0300	FFSR	RO	0x0000000-	32	*Formatter and Flush Status Register, FFSR* on page 9-484
0x0304	FFCR	RW	0x00000000	32	*Formatter and Flush Control Register, FFCR* on page 9-485
0x0308	PSCR	RW	0x0000000a	32	*Periodic Synchronization Counter Register, PSCR* on page 9-488
0x0ED0	ITATBMDATA0	WO	0x00000000	32	*Integration Test ATB Master Data 0 register, ITATBMDATA0* on page 9-489
0x0ED4	ITATBMCTR2	RO	0x00000000	32	*Integration Test ATB Master Control 2 register, ITATBMCTR2* on page 9-491
0x0ED8	ITATBMCTR1	WO	0x00000000	32	*Integration Test ATB Master Control 1 register, ITATBMCTR1* on page 9-492
0x0EE0	ITEVTINTR	WO	0x00000000	32	*Integration Test Event and Interrupt Control Register, ITEVTINTR* on page 9-493

Table 9-333 css600_tmc_etf - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0EE8	ITTRFLIN	RO	0x00000000	32	*Integration Test Trigger In and Flush In register, ITTRFLIN* on page 9-494
0x0EEC	ITATBDATA0	RO	0x00000000	32	*Integration Test ATB Data 0 Register, ITATBDATA0* on page 9-495
0x0EF0	ITATBCTR2	WO	0x00000000	32	*Integration Test ATB Control 2 Register, ITATBCTR2* on page 9-497
0x0EF4	ITATBCTR1	RO	0x00000000	32	*Integration Test ATB Control 1 Register, ITATBCTR1* on page 9-498
0x0EF8	ITATBCTR0	RO	0x00000000	32	*Integration Test ATB Control 0 Register, ITATBCTR0* on page 9-499
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-500
0x0FA0	CLAIMSET	RW	0x0000000f	32	*Claim Tag Set Register, CLAIMSET* on page 9-501
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-502
0x0FB8	AUTHSTATUS	RO	0x00000000	32	*Authentication Status Register, AUTHSTATUS* on page 9-503
0x0FC8	DEVID	RO	0x00000-80	32	*Device Configuration Register, DEVID* on page 9-505
0x0FC4	DEVID1	RO	0x00000001	32	*Device Configuration Register, DEVID1* on page 9-506
0x0FCC	DEVTYPE	RO	0x00000032	32	*Device Type Identifier Register, DEVTYPE* on page 9-507
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-508
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-509
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-510
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-511
0x0FE0	PIDR0	RO	0x000000ea	32	*Peripheral Identification Register 0, PIDR0* on page 9-512
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-513
0x0FE8	PIDR2	RO	0x00000001	32	*Peripheral Identification Register 2, PIDR2* on page 9-514
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-515
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-516
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-517
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-518
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-519

9.13.2 Register descriptions

This section describes the `css600_tmc_etf` registers.

9.13.1 Register summary on page 9-470 provides cross references to individual registers.

RAM Size register, RSZ

Defines the size of trace memory in units of 32-bit words.

The RSZ register characteristics are:

Attributes

Offset `0x0004`

Type Read-write

Reset `0x------00`

Width 32

The following figure shows the bit assignments.

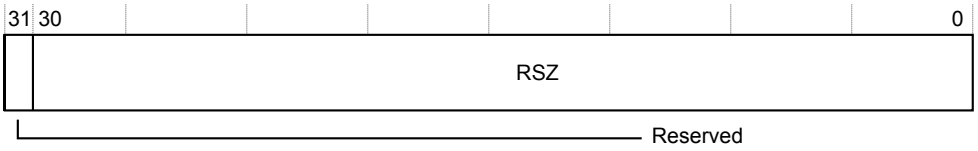

Figure 9-322 RSZ register bit assignments

The following table shows the bit assignments.

Table 9-334 RSZ register bit assignments

Bits	Reset value	Name	Function
[30:0]	IMPLEMENTATION_DEFINED	RSZ	RAM size. Indicates the size of the RAM in 32-bit words. Returns the programmed value on reading. For example: Returns `0x00000100` if trace memory size is 1KB, `0x40000000` if trace memory size is 4GB. This field has the same value as the `MEM_SIZE`parameter.

Status register, STS

Indicates the status of the Trace Memory Controller. After a reset, software must ignore all the fields of this register except STS.TMCReady. The other fields have meaning only when the TMC has left the Disabled state. Writes to all RO fields of this register are ignored.

The STS register characteristics are:

Attributes

Offset	`0x000c`
Type	Read-write
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

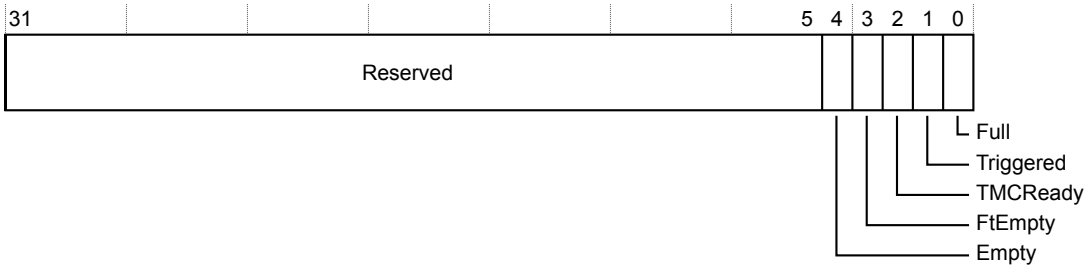

Figure 9-323 STS register bit assignments

The following table shows the bit assignments.

Table 9-335 STS register bit assignments

Bits	Reset value	Name	Function
[4]	UNKNOWN	Empty	Trace buffer empty. If set, this bit indicates that the trace memory does not contain any valid trace data. However, this does not mean that the pipeline stages within the TMC are empty. To determine whether the internal pipeline stages are empty, the software must read the STS.TMCReady bit.
[3]	UNKNOWN	FtEmpty	Formatter pipeline empty. This bit is deprecated and is present in this register to support backwards compatibility with earlier versions of the ETB. It is set when trace capture has stopped, and all internal pipelines and buffers have drained. Unlike STS.TMCReady, it is not affected by buffer drains and AXI accesses. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.
[2]	`0b1`	TMCReady	TMC ready. This flag is set when trace capture has stopped and all internal pipelines and buffers have drained, the TMC is not draining as a result of FFCR.DrainBuffer bit being set (ETF only), and the AXI interface is not busy and the response for final AXI write has been received (ETR only) are all true. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.
[1]	UNKNOWN	Triggered	TMC triggered. This bit is set when trace capture is in progress and the TMC has detected a trigger event. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. A trigger event is when the TMC has written a set number of data words, as programmed in the TRG register, into the trace memory after a rising edge of **trigin** input, or a trigger packet (**atid_s** = `0x7D`) is received in the input trace.
[0]	UNKNOWN	Full	Trace memory full. This bit helps in determining the amount of valid data present in the trace memory. It is not affected by the reprogramming of pointer registers in Disabled state. It is cleared when the TMC leaves Disabled state.

RAM Read Data register, RRD

Reading this register allows data to be read from the trace memory. When the memory width is greater than the APB width (32-bit), multiple reads to the same location must be performed to get the complete memory dataword. The RRP register is incremented only after the last of the required number of read accesses to fetch one memory dataword completes. When the trace memory is empty, this register returns `0xFFFFFFFF`. This value is chosen because it cannot be generated as part of trace data when the formatter is enabled.

The RRD register characteristics are:

Attributes

Offset `0x0010`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

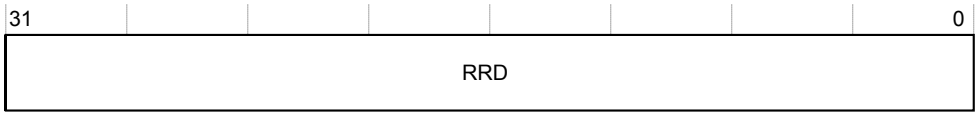

Figure 9-324 RRD register bit assignments

The following table shows the bit assignments.

Table 9-336 RRD register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RRD	Returns the data read from trace memory.

RAM Read Pointer register, RRP

The RAM Read Pointer Register contains the value of the read pointer that is used to read entries from trace memory over the APB interface. Software must program it before enabling trace capture.

The RRP register characteristics are:

Attributes

Offset `0x0014`
Type Read-write
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

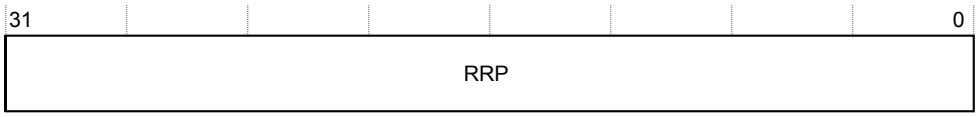

Figure 9-325 RRP register bit assignments

The following table shows the bit assignments.

Table 9-337 RRP register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RRP	The RRP width depends on the size of trace memory and is given by $\log_2$(`MEM_SIZE`x 4). The remaining MSBs of the 32-bit register are of type RAZ/WI.

RAM Write Pointer register, RWP

RAM Write Pointer Register sets the write pointer that is used to write entries into the trace memory. Software must program it before enabling trace capture.

The RWP register characteristics are:

Attributes

Offset `0x0018`

Type Read-write

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

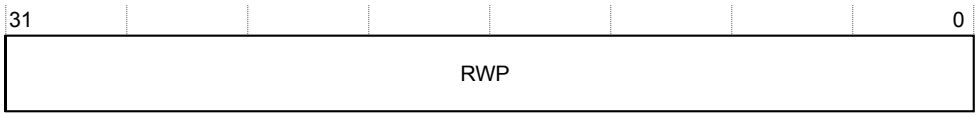

Figure 9-326 RWP register bit assignments

The following table shows the bit assignments.

Table 9-338 RWP register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RWP	The RWP width depends on the size of trace memory and is given by $\log_2$(`MEM_SIZE`x 4). The remaining MSBs of the 32-bit register are of type RAZ/WI.

Trigger Counter register, TRG

In Circular Buffer mode, the Trigger Counter register specifies the number of 32-bit words to capture in the trace memory, after detection of either a rising edge on the **trigin** input or a trigger packet in the incoming trace stream, that is, where **atid_s** = `0x7D`. The value programmed must be aligned to the frame length of 128 bits.

The TRG register characteristics are:

Attributes

Offset	`0x001c`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

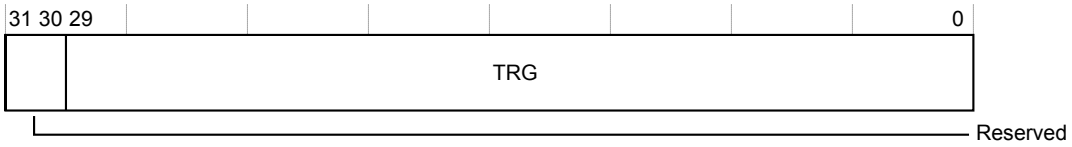

Figure 9-327 TRG register bit assignments

The following table shows the bit assignments.

Table 9-339 TRG register bit assignments

Bits	Reset value	Name	Function
[29:0]	UNKNOWN	TRG	Trigger count. This count represents the number of 32-bit words of trace that are captured between a trigger packet and a trigger event.

Control Register, CTL

This register controls trace stream capture. Setting the CTL.TraceCaptEn bit to 1 enables the TMC to capture the trace data. When trace capture is enabled, the formatter behavior is controlled by the FFCR register.

The CTL register characteristics are:

Attributes

Offset `0x0020`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

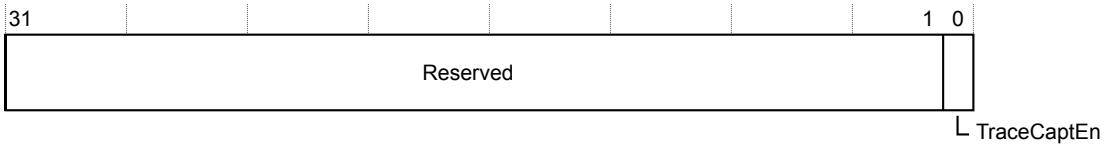

Figure 9-328 CTL register bit assignments

The following table shows the bit assignments.

Table 9-340 CTL register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	TraceCaptEn	Trace capture enable. **0** Disable trace capture. **1** Enable trace capture.

RAM Write Data register, RWD

The RAM Write Data register enables testing of trace memory connectivity to the TMC. When in Disabled state, a write to this register stores data at the location pointed to by the RWP/RWPHI registers. When not in Disabled state, writes are ignored. When the memory data width, as indicated by the DEVID.MEM_WIDTH register field, is greater than 32 bits, multiple writes to this register must be performed together to write a full memory width of data. For example, if the memory width is 128 bits, then writes to this register must be performed 4 at a time. When a full memory width of data has been written to this register, the data is written to the trace memory and the RWP is incremented to the next memory word.

The RWD register characteristics are:

Attributes

Offset `0x0024`

Type Write-only

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

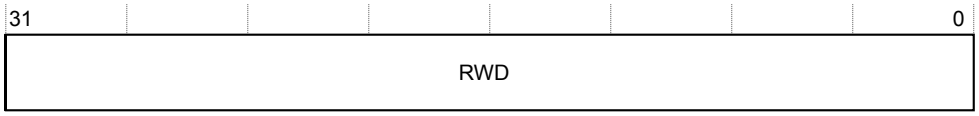

Figure 9-329 RWD register bit assignments

The following table shows the bit assignments.

Table 9-341 RWD register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	RWD	Data written to this register is placed in the trace memory.

Mode register, MODE

This register controls the TMC operating mode. The operating mode can only be changed when the TMC is in Disabled state, that is, when CTL.TraceCaptEn = 0. Attempting to write to this register in any other state results in UNPREDICTABLE behavior. The operating mode is ignored when in Disabled state.

The MODE register characteristics are:

Attributes

Offset `0x0028`
Type Read-write
Reset `0x0000000-`
Width 32

The following figure shows the bit assignments.

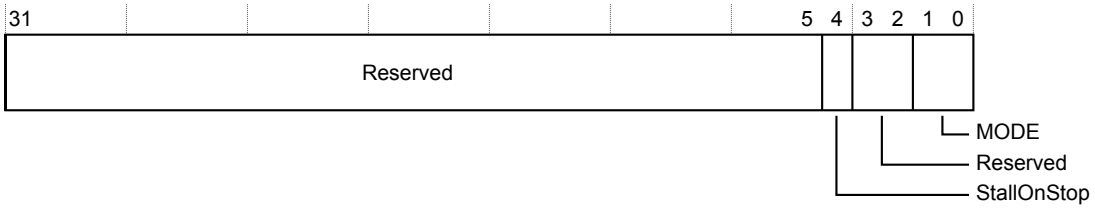

Figure 9-330 MODE register bit assignments

The following table shows the bit assignments.

Table 9-342 MODE register bit assignments

Bits	Reset value	Name	Function
[4]	`0b0`	StallOnStop	Stall On Stop. If this bit is set and the formatter stops as a result of a stop event, the output **atready_s** is de-asserted to stall the ATB interface and avoid loss of trace. If this bit is clear and the formatter stops as a result of a stop event, signal **atready_s** remains asserted but the TMC discards further incoming trace.
[1:0]	UNKNOWN	MODE	Selects the operating mode. If a reserved MODE value is programmed and trace capture is enabled, the TMC starts to operate in SWF1 mode. However, reading the MODE.MODE field returns the programmed value. `0x0` CB, Circular Buffer mode. `0x1` SWF1, Software Read FIFO mode 1. `0x2` HWF, Hardware Read FIFO mode. `0x3` Reserved.

Latched Buffer Fill Level, LBUFLEVEL

Reading this register returns the maximum fill level of the trace memory in 32-bit words since this register was last read. Reading this register also results in its contents being updated to the current fill level. When entering Disabled state, it retains its last value. While in Disabled state, reads from this register do not affect its value. When exiting Disabled state, the LBUFLEVEL register is updated to the current fill level.

The LBUFLEVEL register characteristics are:

Attributes

Offset `0x002c`
Type Read-only
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

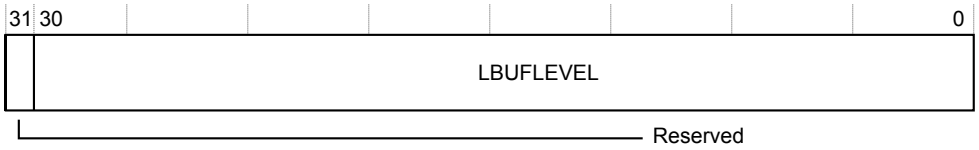

Figure 9-331 LBUFLEVEL register bit assignments

The following table shows the bit assignments.

Table 9-343 LBUFLEVEL register bit assignments

Bits	Reset value	Name	Function
[30:0]	UNKNOWN	LBUFLEVEL	Latched Buffer Fill Level. Indicates the maximum fill level of the trace memory in 32-bit words since this register was last read. The width of the register is 1 + $\log_2$(`MEM_SIZE`).

Current Buffer Fill Level, CBUFLEVEL

The CBUFLEVEL register indicates the current fill level of the trace memory in units of 32-bit words. When the TMC leaves Disabled state, this register dynamically indicates the current fill level of trace memory. It retains its value on entering Disabled state. It is not affected by the reprogramming of pointer registers in Disabled state. However, RRD reads when using the Circular Buffer in Disabled state are a special case. In this case the register indicates the updated fill level on RRD reads.

The CBUFLEVEL register characteristics are:

Attributes

Offset `0x0030`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

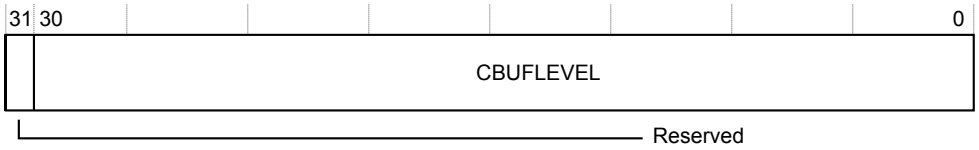

Figure 9-332 CBUFLEVEL register bit assignments

The following table shows the bit assignments.

Table 9-344 CBUFLEVEL register bit assignments

Bits	Reset value	Name	Function
[30:0]	UNKNOWN	CBUFLEVEL	Current Buffer Fill Level. Indicates the current fill level of the trace memory in 32-bit words. The width of the register is $1 + \log_2$(`MEM_SIZE`).

Buffer Level Water Mark, BUFWM

The value that is programmed into this register indicates the desired threshold vacancy level in 32-bit words in the trace memory. When the available space in the FIFO is less than or equal to this value, that is, fill level >= (MEM_SIZE- BUFWM), the **full** output is asserted and the STS.Full bit is set. This register is used only in the FIFO modes, that is, SWF1, SWF2, and HWF modes. In CB mode, the same functionality is obtained by programming the RWP to the desired vacancy trigger level, so that when the pointer wraps around, the **full** output gets asserted indicating that the vacancy level has fallen below the desired level. Reading this register returns the programmed value. The maximum value that can be written into this register is MEM_SIZE- 1, in which case the **full** output is asserted after the first 32-bit word is written to trace memory. Writing to this register other than when in Disabled state results in UNPREDICTABLE behavior. Any software using it must program it with an initial value before setting the CTL.TraceCaptEn bit to 1.

The BUFWM register characteristics are:

Attributes

Offset	0x0034
Type	Read-write
Reset	0x--------
Width	32

The following figure shows the bit assignments.

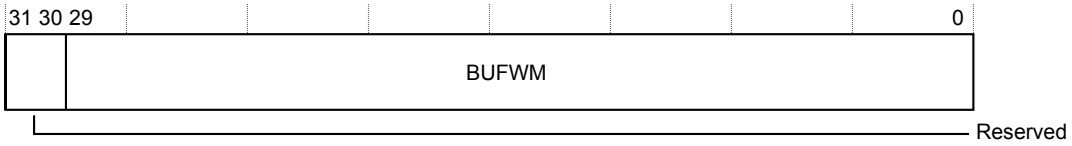

Figure 9-333 BUFWM register bit assignments

The following table shows the bit assignments.

Table 9-345 BUFWM register bit assignments

Bits	Reset value	Name	Function
[29:0]	UNKNOWN	BUFWM	Buffer Level Watermark. Indicates the desired threshold vacancy level in 32-bit words in the trace memory. The width of the register is $\log_2$(MEM_SIZE).

Formatter and Flush Status Register, FFSR

This register indicates the status of the Formatter, and the status of Flush request.

The FFSR register characteristics are:

Attributes

Offset	`0x0300`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

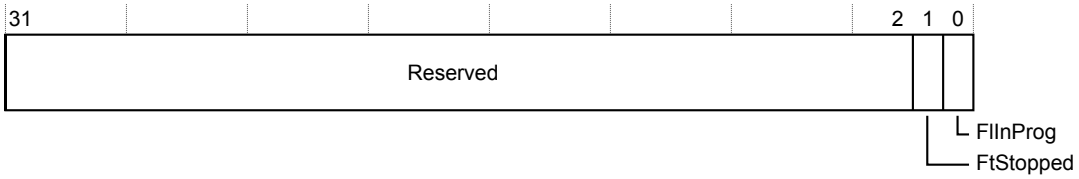

Figure 9-334 FFSR register bit assignments

The following table shows the bit assignments.

Table 9-346 FFSR register bit assignments

Bits	Reset value	Name	Function
[1]	UNKNOWN	FtStopped	Formatter Stopped. This bit behaves the same way as STS.FtEmpty. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. The FFCR.FtStopped bit is deprecated and is present in this register to support backwards-compatibility with earlier versions of the ETB. **0** Trace capture has not yet completed. **1** Trace capture has completed and all captured trace data has been written to the trace memory.
[0]	UNKNOWN	FlInProg	Flush In Progress. This bit indicates whether the TMC is currently processing a flush request. In the ETB and ETR configurations, the flush initiation is controlled by the flush control bits in the FFCR register. In the ETF configuration, the flush request can also be from the ATB Master port. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. When in Disabled state, this bit is not updated. **0** No flush activity in progress. **1** Flush in progress on the ATB slave interface or the TMC internal pipeline.

Formatter and Flush Control Register, FFCR

This register allows user control of the stop, trigger, and flush events.

The FFCR register characteristics are:

Attributes

Offset	`0x0304`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

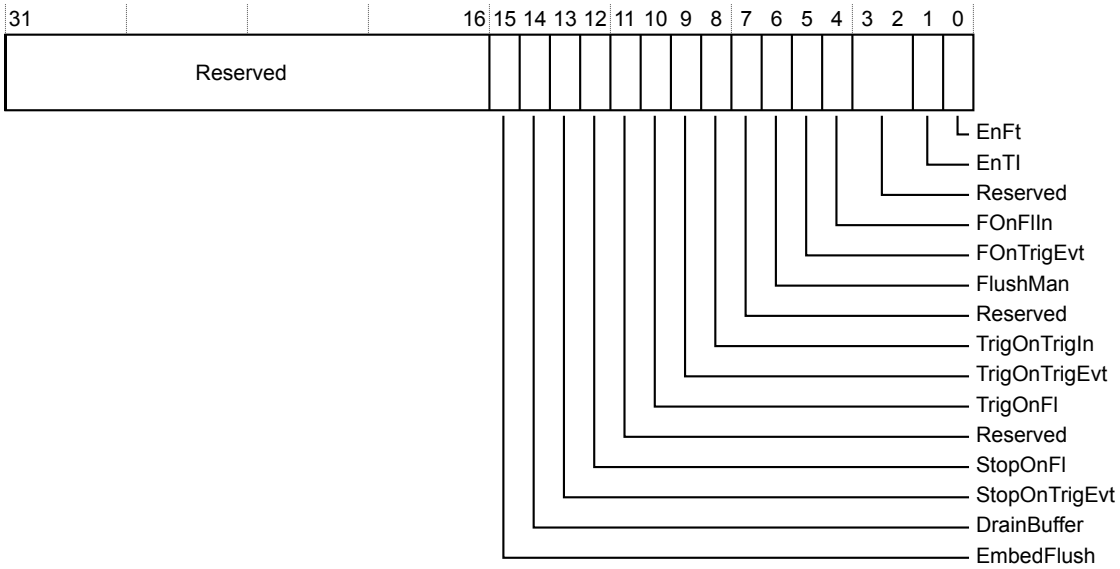

Figure 9-335 FFCR register bit assignments

The following table shows the bit assignments.

Table 9-347 FFCR register bit assignments

Bits	Reset value	Name	Function
[15]	0b0	EmbedFlush	Embed Flush ID (flush completion packet). Enables insertion of Flush ID 0x7B with a single byte of data payload = 0x00 in the output trace, immediately after the last flush data byte, when a flush completes on the ATB slave interface. This bit is effective only in Normal formatting modes. In Bypass mode, the Flush ID insertion remains disabled and this bit is ignored. **0** Disable Flush ID insertion. **1** Enable Flush ID insertion.
[14]	0b0	DrainBuffer	Drain Buffer. This bit is used to enable draining of the trace data through the ATB master interface after the formatter has stopped. It is useful in CB mode to capture trace data into trace memory and then to drain the captured trace through the ATB master interface. Writing a 1 to this bit when in Stopped state starts the drain of the contents of trace buffer. This bit always reads as 0. The STS.TMCReady bit goes LOW while the drain is in progress. This bit is functional only when the TMC is in CB mode and formatting is enabled, that is, FFCR.EnFt=1. Setting this bit when the TMC is in any other mode, or when not in Stopped state, results in UNPREDICTABLE behavior. When trace capture is complete in CB mode, all of the captured trace must be retrieved from the trace memory, either by reading all trace data out through RRD reads, or draining all trace data by setting the FFCR.DrainBuffer bit. Setting this bit after some of the captured trace has been read out through RRD results in UNPREDICTABLE behavior.
[13]	0b0	StopOnTrigEvt	Stop On Trigger Event. If this bit is set, the formatter is stopped when a Trigger Event has been observed. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[12]	0b0	StopOnFl	Stop On Flush. If this bit is set, the formatter is stopped on completion of a flush operation. The initiation of a flush operation is controlled by programming the register bits FFCR.FlushMan, FFCR.FOnTrigEvt, and FFCR.FOnFlIn. When a flush-initiation condition occurs, **afvalid_s** is asserted, and when the flush completion is received, that is, **afready_s**=1, trace capture is stopped. Any remaining data in the formatter is appended with a post-amble and written to trace memory. The flush operation is then complete. When the TMC is configured as an ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a formatter stop regardless of the value that is programmed in this bit.
[10]	0b0	TrigOnFl	Indicate on trace stream the completion of flush. If this bit is set, a trigger is indicated on the trace stream when **afready_s** is received for a flush in progress. If this bit is clear, no triggers are embedded in the trace stream on flush completion. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. When the TMC is configured as ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a trigger indication on the trace stream regardless of the value that is programmed in this bit.
[9]	0b0	TrigOnTrigEvt	Indicate on trace stream the occurrence of a Trigger Event. If this bit is set, a trigger is indicated on the output trace stream when a Trigger Event occurs. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[8]	0b0	TrigOnTrigIn	Indicate on trace stream the occurrence of a rising edge on **trigin**. If this bit is set, a trigger is indicated on the trace stream when a rising edge is detected on the **trigin** input. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit.

Table 9-347 FFCR register bit assignments (continued)

Bits	Reset value	Name	Function
[6]	0b0	FlushMan	Manually generate a flush of the system. Writing 1 to this bit causes a flush to be generated. This bit is cleared automatically when, in formatter bypass mode, **afready_s** was sampled high, or, in normal formatting mode, **afready_s** was sampled high and all flush data was output to the trace memory. If CTL.TraceCaptEn=0, writes to this bit are ignored.
[5]	0b0	FOnTrigEvt	Flush on Trigger Event. Setting this bit generates a flush when a Trigger Event occurs. If FFCR.StopOnTrigEvt is set, this bit is ignored. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[4]	0b0	FOnFlIn	Setting this bit enables the detection of transitions on the **flushin** input by the TMC. If this bit is set and the formatter has not already stopped, a rising edge on **flushin** initiates a flush request.
[1]	0b0	EnTI	Enable Trigger Insertion. Setting this bit enables the insertion of triggers in the formatted trace stream. A trigger is indicated by inserting one byte of data 0x00 with **atid_s**=0x7D in the trace stream. Trigger indication on the trace stream is also controlled by the register bits FFCR.TrigOnFl, FFCR.TrigOnTrigEvt, and FFCR.TrigOnTrigIn. This bit can only be changed when the TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled.
[0]	0b0	EnFt	Enable Formatter. If this bit is set, formatting is enabled. This bit takes effect when not in Disabled state, and is ignored in Disabled state. If this bit is clear, formatting is disabled. In this case, the incoming trace data is assumed to be from a single trace source. If multiple trace IDs are received by the TMC when trace capture is enabled and the formatter is disabled, it results in interleaving of trace data. Disabling of formatting is deprecated, and is supported in the TMC for backwards-compatibility with earlier versions of the ETB. Therefore, disabling of formatting is supported only in CB mode. Features in the TMC such as the FIFO modes and the FFCR.DrainBuffer bit that are not part of the earlier versions of the ETB do not support disabling of formatting. This bit can only be changed when TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled. If the TMC is enabled in a mode other than Circular Buffer mode with formatting disabled, it results in formatting being enabled.

Periodic Synchronization Counter Register, PSCR

This register determines the reload value of the Periodic Synchronization Counter. This counter enables the frequency of sync packets to be optimized to the trace capture buffer size. The default behavior of the counter is to generate periodic synchronization requests, **syncreq_s**, on the ATB slave interface.

The PSCR register characteristics are:

Attributes

Offset `0x0308`
Type Read-write
Reset `0x0000000a`
Width 32

The following figure shows the bit assignments.

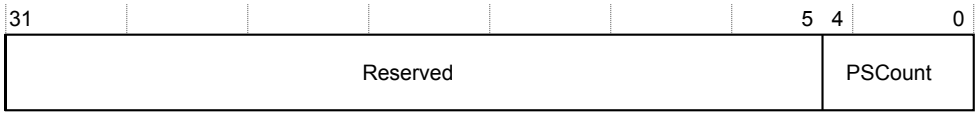

Figure 9-336 PSCR register bit assignments

The following table shows the bit assignments.

Table 9-348 PSCR register bit assignments

Bits	Reset value	Name	Function
[4:0]	`0b01010`	PSCount	Periodic Synchronization Count. Determines the reload value of the Synchronization Counter. The reload value takes effect the next time the counter reaches zero. When trace capture is enabled, the Synchronization Counter counts the number of bytes of trace data that is stored into the trace memory, regardless of whether the trace data has been formatted by the TMC or not, since the occurrence of the last sync request on the ATB slave interface. When the counter reaches 0, a sync request is sent on the ATB slave interface. Reads from this register return the reload value that is programmed in this register. This field resets to `0x0A`, that is, the default sync period is 2^10 bytes. If a reserved value is programmed in this register field, the value `0x1B` is used instead, and subsequent reads from this register also return `0x1B`. The following constraints apply to the values written to the PSCount field: `0x0` - synchronization is disabled, `0x1-0x6` - reserved, `0x7-0x1B` - synchronization period is 2^PSCount bytes. The smallest value `0x7` gives a sync period of 128 bytes. The maximum allowed value `0x1B` gives a sync period of 2^27 bytes, `0x1C-0x1F` - reserved.

Integration Test ATB Master Data 0 register, ITATBMDATA0

This register enables control of the **atdata_m** output in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any given bit is driven on the output pin that is controlled by that bit and the reads return `0x0`. The width of this register is given by 1+(ATB DATA WIDTH)/8.

The ITATBMDATA0 register characteristics are:

Attributes

Offset	`0x0ed0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

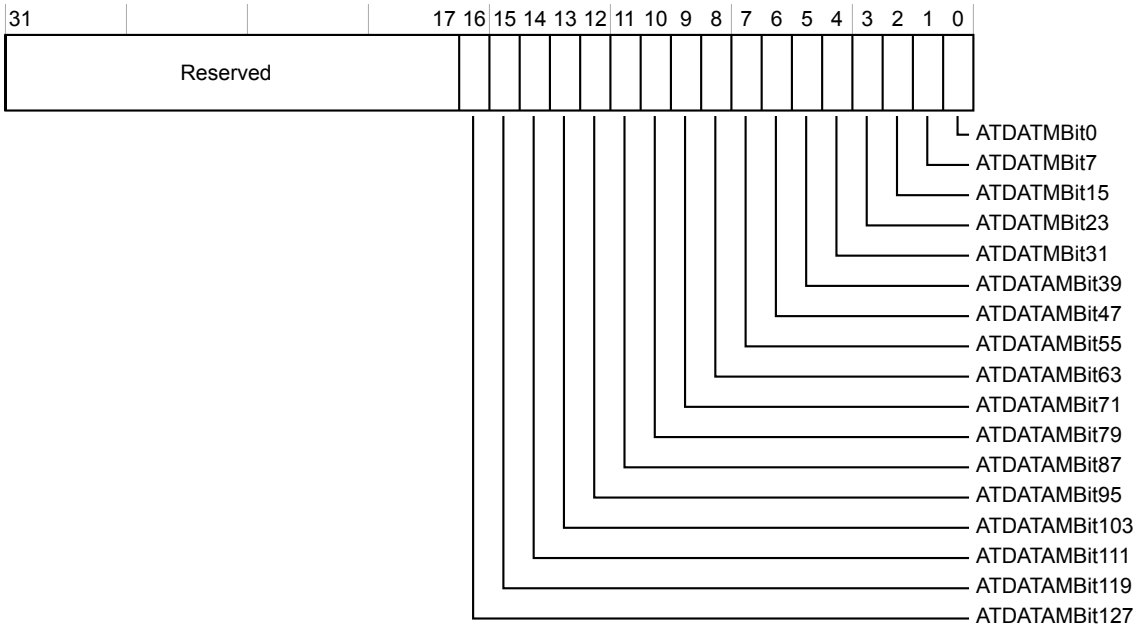

Figure 9-337 ITATBMDATA0 register bit assignments

The following table shows the bit assignments.

Table 9-349 ITATBMDATA0 register bit assignments

Bits	Reset value	Name	Function
[16]	`0b0`	ATDATAMBit127	Controls the value of **atdata_m[127]** output in integration mode.
[15]	`0b0`	ATDATAMBit119	Controls the value of **atdata_m[119]** output in integration mode.
[14]	`0b0`	ATDATAMBit111	Controls the value of **atdata_m[111]** output in integration mode.
[13]	`0b0`	ATDATAMBit103	Controls the value of **atdata_m[103]** output in integration mode.
[12]	`0b0`	ATDATAMBit95	Controls the value of **atdata_m[95]** output in integration mode.
[11]	`0b0`	ATDATAMBit87	Controls the value of **atdata_m[87]** output in integration mode.
[10]	`0b0`	ATDATAMBit79	Controls the value of **atdata_m[79]** output in integration mode.
[9]	`0b0`	ATDATAMBit71	Controls the value of **atdata_m[71]** output in integration mode.

Table 9-349 ITATBMDATA0 register bit assignments (continued)

Bits	Reset value	Name	Function
[8]	0b0	ATDATAMBit63	Controls the value of **atdata_m[63]** output in integration mode.
[7]	0b0	ATDATAMBit55	Controls the value of **atdata_m[55]** output in integration mode.
[6]	0b0	ATDATAMBit47	Controls the value of **atdata_m[47]** output in integration mode.
[5]	0b0	ATDATAMBit39	Controls the value of **atdata_m[39]** output in integration mode.
[4]	0b0	ATDATMBit31	Controls the value of **atdata_m[31]** output in integration mode.
[3]	0b0	ATDATMBit23	Controls the value of **atdata_m[23]** output in integration mode.
[2]	0b0	ATDATMBit15	Controls the value of **atdata_m[15]** output in integration mode.
[1]	0b0	ATDATMBit7	Controls the value of **atdata_m[7]** output in integration mode.
[0]	0b0	ATDATMBit0	Controls the value of **atdata_m[0]** output in integration mode.

Integration Test ATB Master Control 2 register, ITATBMCTR2

This register captures the values of ATB master inputs **atready_m**, **afvalid_m**, and **syncreq_m** in integration mode. In functional mode, this register behaves as RAZ/WI.

The ITATBMCTR2 register characteristics are:

Attributes

Offset `0x0ed4`

Type Read-only

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

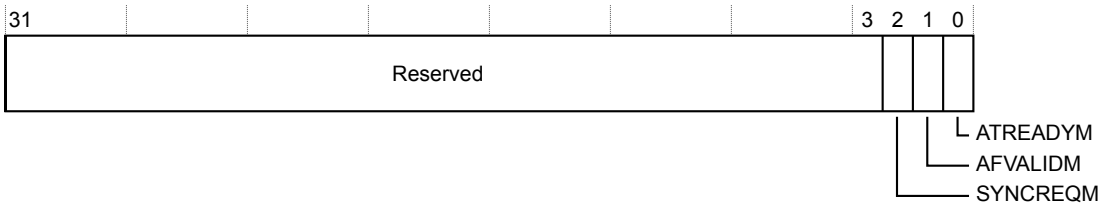

Figure 9-338 ITATBMCTR2 register bit assignments

The following table shows the bit assignments.

Table 9-350 ITATBMCTR2 register bit assignments

Bits	Reset value	Name	Function
[2]	`0b0`	SYNCREQM	Integration status of **syncreq_m** input. In integration mode, this bit latches to 1 on a rising edge of **syncreq_m** input, which is cleared when this register is read or when integration mode is disabled.
[1]	`0b0`	AFVALIDM	Integration status of **afvalid_m** input. In integration mode, writes are ignored and reads return the value of **afvalid_m** input.
[0]	`0b0`	ATREADYM	Integration status of **atready_m** input. In integration mode, writes are ignored and reads return the value of the **atready_m** input.

Integration Test ATB Master Control 1 register, ITATBMCTR1

This register enables control of the **atid_m** output in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to this register is driven on the **atid_m** output and the reads return `0x0`.

The ITATBMCTR1 register characteristics are:

Attributes

Offset	`0x0ed8`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 9-339 ITATBMCTR1 register bit assignments

The following table shows the bit assignments.

Table 9-351 ITATBMCTR1 register bit assignments

Bits	Reset value	Name	Function
[6:0]	`0b0000000`	ATIDM	Controls the value of the **atid_m[6:0]** output in integration mode.

Integration Test Event and Interrupt Control Register, ITEVTINTR

This register controls the values of event and interrupt outputs in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITEVTINTR register characteristics are:

Attributes

Offset	`0x0ee0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

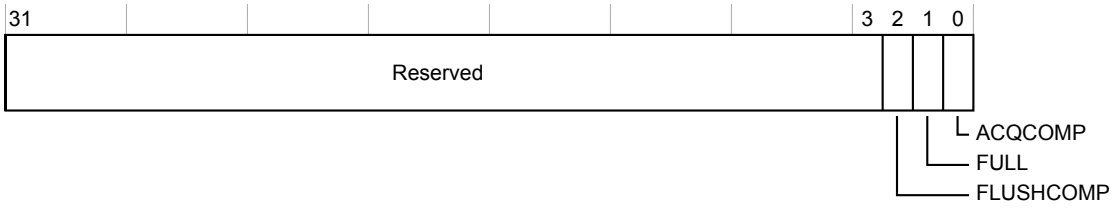

Figure 9-340 ITEVTINTR register bit assignments

The following table shows the bit assignments.

Table 9-352 ITEVTINTR register bit assignments

Bits	Reset value	Name	Function
[2]	`0b0`	FLUSHCOMP	Controls the value of **flushcomp** output in integration mode.
[1]	`0b0`	FULL	Controls the value of **full** output in integration mode.
[0]	`0b0`	ACQCOMP	Controls the value of **acqcomp** output in integration mode.

Integration Test Trigger In and Flush In register, ITTRFLIN

This register captures the values of the **flushin** and **trigin** inputs in integration mode. In functional mode, this register behaves as RAZ/WI.

The ITTRFLIN register characteristics are:

Attributes

Offset	`0x0ee8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

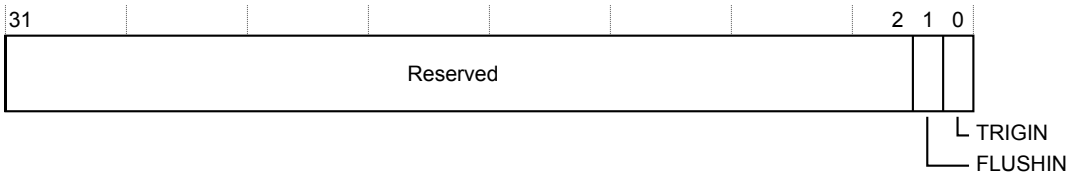

Figure 9-341 ITTRFLIN register bit assignments

The following table shows the bit assignments.

Table 9-353 ITTRFLIN register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	FLUSHIN	Integration status of **flushin** input. In integration mode, this bit latches to 1 on a rising edge of the **flushin** input. It is cleared when the register is read or when integration mode is disabled.
[0]	`0b0`	TRIGIN	Integration status of **trigin** input. In integration mode, this bit latches to 1 on a rising edge of the **trigin** input. It is cleared when the register is read or when integration mode is disabled.

Integration Test ATB Data 0 Register, ITATBDATA0

This register captures the value of **atdata_s** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding **atdata_s** bits. The width of this register is given by: 1+(ATB DATA WIDTH)/8.

The ITATBDATA0 register characteristics are:

Attributes

Offset	`0x0eec`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

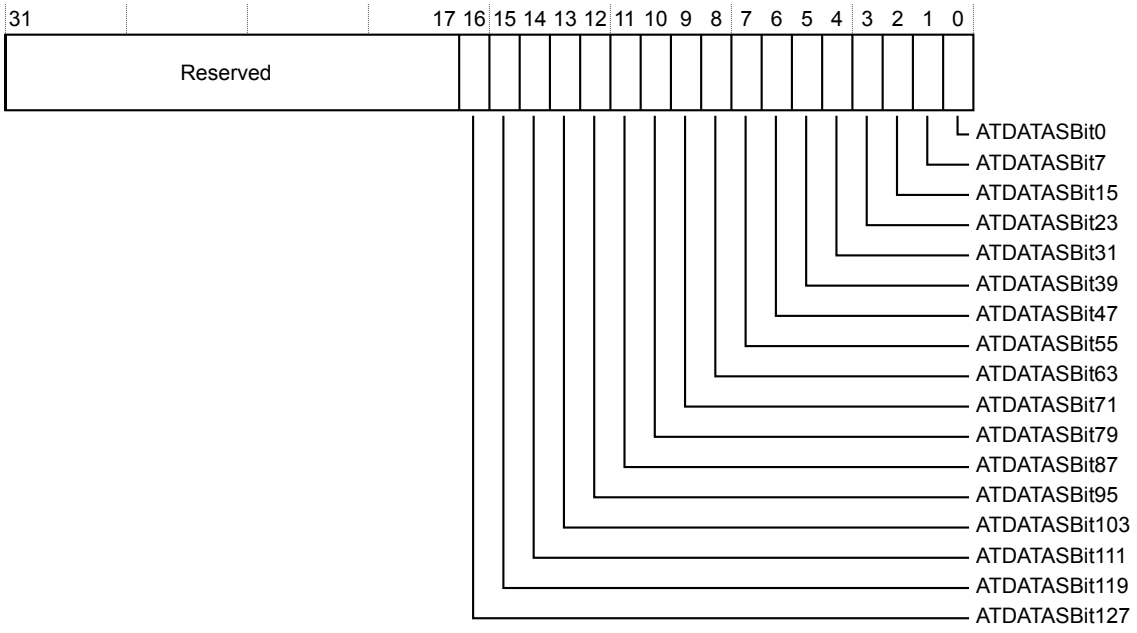

Figure 9-342 ITATBDATA0 register bit assignments

The following table shows the bit assignments.

Table 9-354 ITATBDATA0 register bit assignments

Bits	Reset value	Name	Function
[16]	`0b0`	ATDATASBit127	Returns the value of **atdata_s[127]** input in integration mode.
[15]	`0b0`	ATDATASBit119	Returns the value of **atdata_s[119]** input in integration mode.
[14]	`0b0`	ATDATASBit111	Returns the value of **atdata_s[111]** input in integration mode.
[13]	`0b0`	ATDATASBit103	Returns the value of **atdata_s[103]** input in integration mode.
[12]	`0b0`	ATDATASBit95	Returns the value of **atdata_s[95]** input in integration mode.
[11]	`0b0`	ATDATASBit87	Returns the value of **atdata_s[87]** input in integration mode.
[10]	`0b0`	ATDATASBit79	Returns the value of **atdata_s[79]** input in integration mode.
[9]	`0b0`	ATDATASBit71	Returns the value of **atdata_s[71]** input in integration mode.
[8]	`0b0`	ATDATASBit63	Returns the value of **atdata_s[63]** input in integration mode.

Table 9-354 ITATBDATA0 register bit assignments (continued)

Bits	Reset value	Name	Function
[7]	0b0	ATDATASBit55	Returns the value of **atdata_s[55]** input in integration mode.
[6]	0b0	ATDATASBit47	Returns the value of **atdata_s[47]** input in integration mode.
[5]	0b0	ATDATASBit39	Returns the value of **atdata_s[39]** input in integration mode.
[4]	0b0	ATDATASBit31	Returns the value of **atdata_s[31]** input in integration mode.
[3]	0b0	ATDATASBit23	Returns the value of **atdata_s[23]** input in integration mode.
[2]	0b0	ATDATASBit15	Returns the value of **atdata_s[15]** input in integration mode.
[1]	0b0	ATDATASBit7	Returns the value of **atdata_s[7]** input in integration mode.
[0]	0b0	ATDATASBit0	Returns the value of **atdata_s[0]** input in integration mode.

Integration Test ATB Control 2 Register, ITATBCTR2

This register enables control of ATB slave outputs **atready_s**, **afvalid_s**, and **syncreq_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITATBCTR2 register characteristics are:

Attributes

Offset	`0x0ef0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

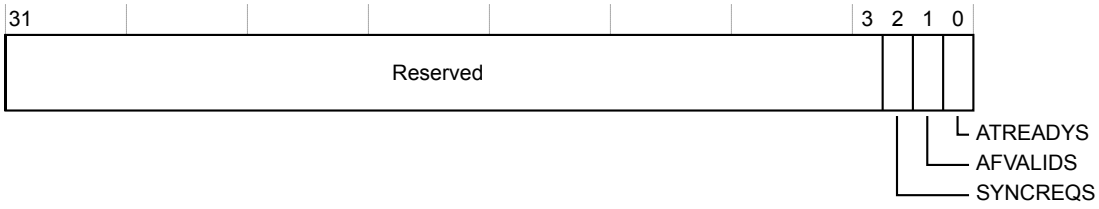

Figure 9-343 ITATBCTR2 register bit assignments

The following table shows the bit assignments.

Table 9-355 ITATBCTR2 register bit assignments

Bits	Reset value	Name	Function
[2]	`0b0`	SYNCREQS	Controls the value of **syncreq_s** output in integration mode.
[1]	`0b0`	AFVALIDS	Controls the value of **afvalid_s** output in integration mode.
[0]	`0b0`	ATREADYS	Controls the value of **atready_s** output in integration mode.

Integration Test ATB Control 1 Register, ITATBCTR1

This register captures the value of the **atid_s[6:0]** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of **atid_s** input.

The ITATBCTR1 register characteristics are:

Attributes

Offset	`0x0ef4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

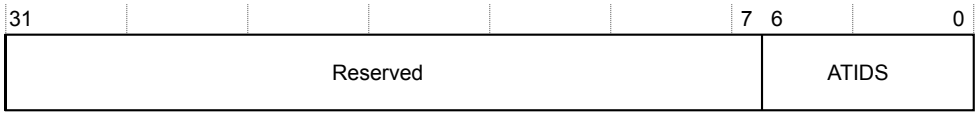

Figure 9-344 ITATBCTR1 register bit assignments

The following table shows the bit assignments.

Table 9-356 ITATBCTR1 register bit assignments

Bits	Reset value	Name	Function
[6:0]	`0b0000000`	ATIDS	Returns the value of **atid_s[6:0]** input in integration mode.

Integration Test ATB Control 0 Register, ITATBCTR0

This register captures the values of ATB slave inputs **atvalid_s**, **afready_s**, **atwakeup_s**, and **atbytes_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding input pins. The width of this register is given by: $8+\log_2$(ATB DATA WIDTH/8).

The ITATBCTR0 register characteristics are:

Attributes

Offset	`0x0ef8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

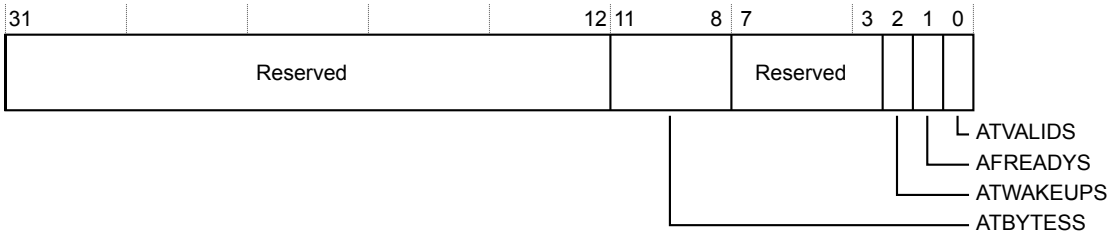

Figure 9-345 ITATBCTR0 register bit assignments

The following table shows the bit assignments.

Table 9-357 ITATBCTR0 register bit assignments

Bits	Reset value	Name	Function
[11:8]	`0b0000`	ATBYTESS	Returns the value of **atbytes_s** input in integration mode. N=8+$\log_2$(ATB DATA WIDTH/8).
[2]	`0b0`	ATWAKEUPS	Returns the value of **atwakeup_s** input in integration mode.
[1]	`0b0`	AFREADYS	Returns the value of **afready_s** input in integration mode.
[0]	`0b0`	ATVALIDS	Returns the value of **atvalid_s** input in integration mode.

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.

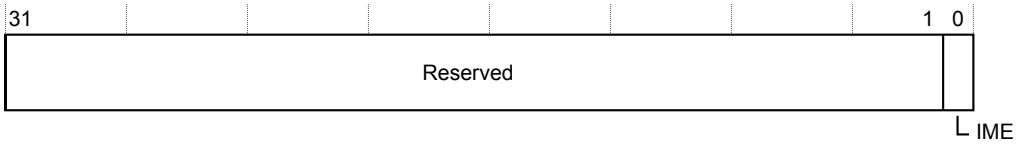

Figure 9-346 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-358 ITCTRL register bit assignments

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

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 `0x0000000f`
Width 32

The following figure shows the bit assignments.

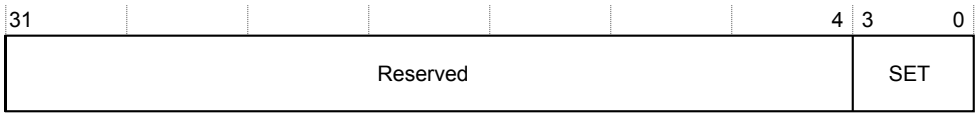

Figure 9-347 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-359 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	SET	A bit programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

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.

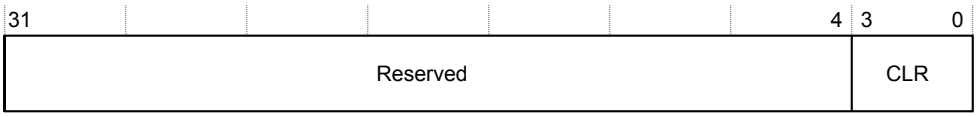

Figure 9-348 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-360 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	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.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

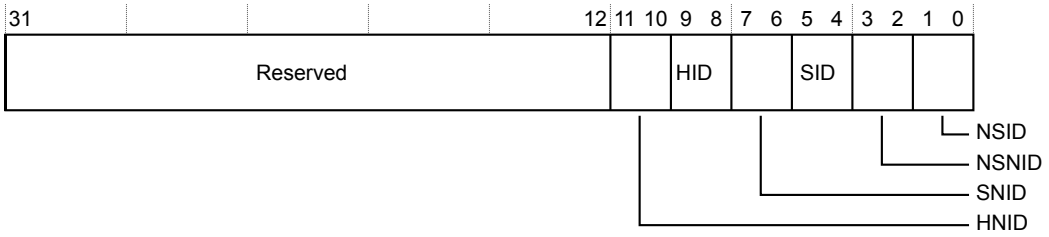

Figure 9-349 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-361 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	`0b00`	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-361 AUTHSTATUS register bit assignments (continued)

Bits	Reset value	Name	Function
[3:2]	0b00	NSNID	Non-secure non-invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.
[1:0]	0b00	NSID	Non-secure invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.

Device Configuration Register, DEVID

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

The DEVID register characteristics are:

Attributes

Offset `0x0fc8`
Type Read-only
Reset `0x00000-80`
Width 32

The following figure shows the bit assignments.

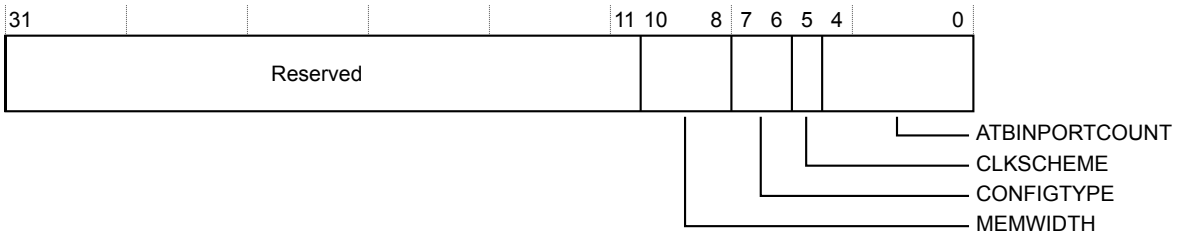

Figure 9-350 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-362 DEVID register bit assignments

Bits	Reset value	Name	Function
[10:8]	IMPLEMENTATION_DEFINED	MEMWIDTH	This value is twice `ATB_DATA_WIDTH`. `0x2` Memory interface databus is 32-bits wide. `0x3` Memory interface databus is 64-bits wide. `0x4` Memory interface databus is 128-bits wide. `0x5` Memory interface databus is 256-bits wide.
[7:6]	`0b10`	CONFIGTYPE	Returns `0x2`, indicating ETF configuration.
[5]	`0b0`	CLKSCHEME	RAM Clocking Scheme. This value indicates the TMC RAM clocking scheme used, that is, whether the TMC RAM operates synchronously or asynchronously to the TMC clock. Fixed to 0 indicating that TMC RAM clock is synchronous to the **clk** input.
[4:0]	`0b00000`	ATBINPORTCOUNT	Hidden Level of ATB input multiplexing. This value indicates the type/number of ATB multiplexing present on the input ATB. Fixed to `0x00` indicating that no multiplexing is present.

Device Configuration Register, DEVID1

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

The DEVID1 register characteristics are:

Attributes

Offset `0x0fc4`

Type Read-only

Reset `0x00000001`

Width 32

The following figure shows the bit assignments.

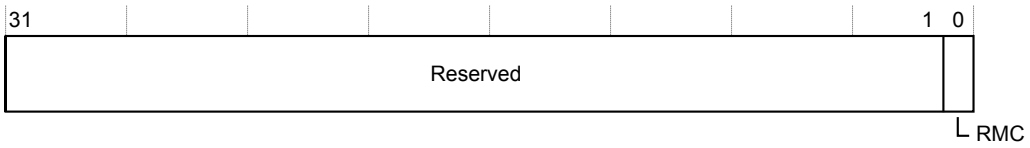

Figure 9-351 DEVID1 register bit assignments

The following table shows the bit assignments.

Table 9-363 DEVID1 register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x00000000`	-	Reserved.
[0]	1	RMC	Indicates register management mode 1 is implemented.

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 `0x00000032`
Width 32

The following figure shows the bit assignments.

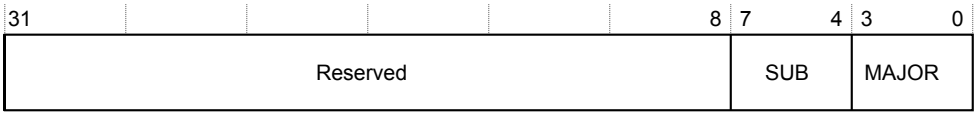

Figure 9-352 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-364 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0011`	SUB	Minor classification. Returns `0x3`, indicating this component is a FIFO.
[3:0]	`0b0010`	MAJOR	Major classification. Returns `0x2`, indicating this component is a Trace Link.

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.

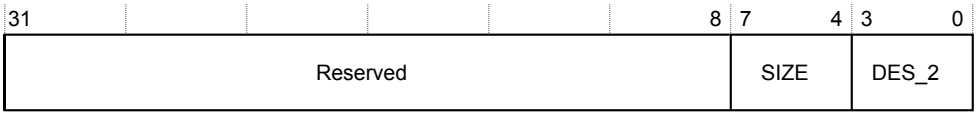

Figure 9-353 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-365 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

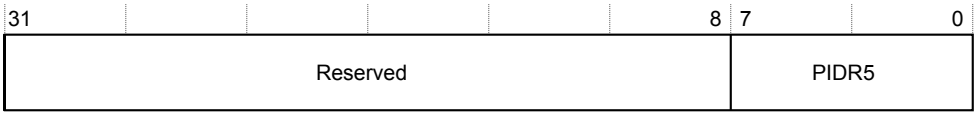

Figure 9-354 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-366 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

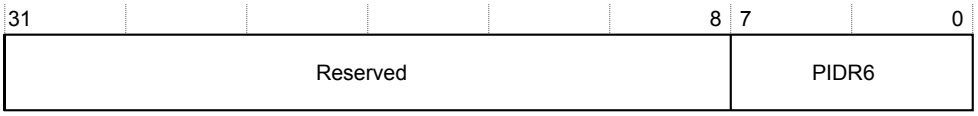

Figure 9-355 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-367 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

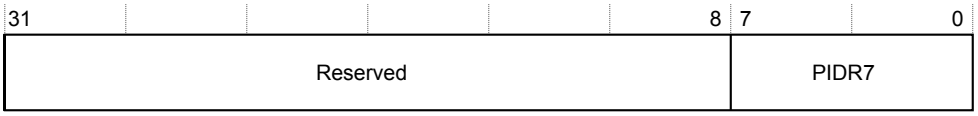

Figure 9-356 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-368 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000ea`

Width 32

The following figure shows the bit assignments.

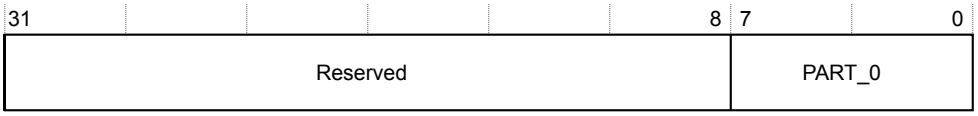

Figure 9-357 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-369 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11101010`	PART_0	Part number (lower 8 bits). Returns `0xea`, indicating TMC ETF.

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.

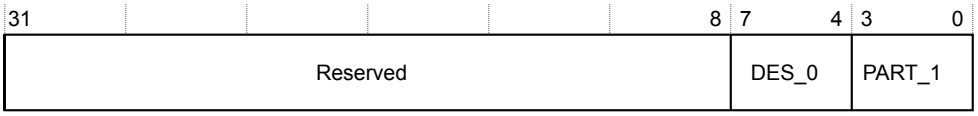

Figure 9-358 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-370 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x00000001`
Width 32

The following figure shows the bit assignments.

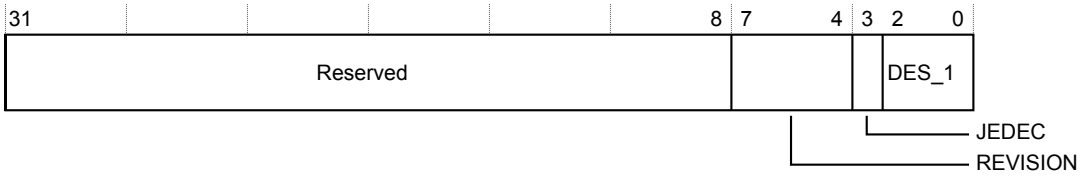

Figure 9-359 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-371 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0x1`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

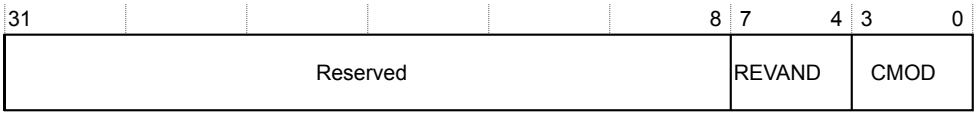

Figure 9-360 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-372 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

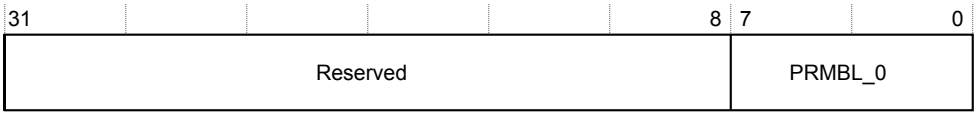

Figure 9-361 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-373 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

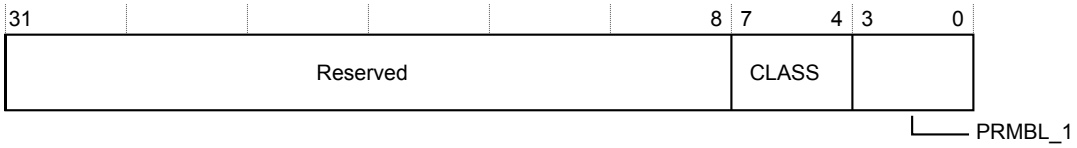

Figure 9-362 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-374 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

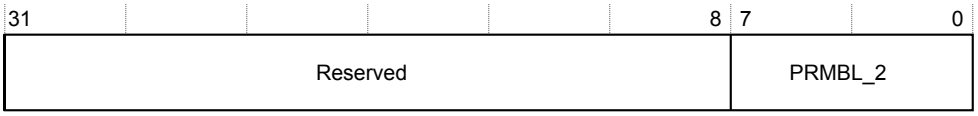

Figure 9-363 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-375 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

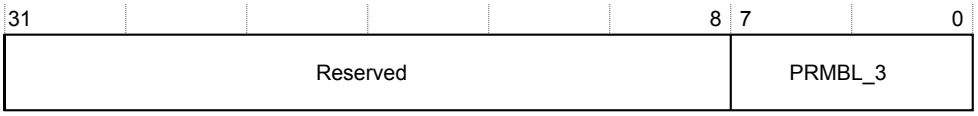

Figure 9-364 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-376 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.14 css600_tmc_etr introduction

This section describes the functions and programmers model of the `css600_tmc_etr`.

This section contains the following subsections:

- *9.14.1 Register summary* on page 9-520.
- *9.14.2 Register descriptions* on page 9-521.

9.14.1 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.

Table 9-377 css600_tmc_etr - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0004	RSZ	RW	0x--------	32	*RAM Size register, RSZ* on page 9-522
0x000C	STS	RW	0x000000--	32	*Status register, STS* on page 9-523
0x0010	RRD	RO	0x--------	32	*RAM Read Data register, RRD* on page 9-525
0x0014	RRP	RW	0x--------	32	*RAM Read Pointer register, RRP* on page 9-526
0x0018	RWP	RW	0x--------	32	*RAM Write Pointer register, RWP* on page 9-527
0x001C	TRG	RW	0x--------	32	*Trigger Counter register, TRG* on page 9-528
0x0020	CTL	RW	0x00000000	32	*Control Register, CTL* on page 9-529
0x0024	RWD	WO	0x00000000	32	*RAM Write Data register, RWD* on page 9-530
0x0028	MODE	RW	0x0000000-	32	*Mode register, MODE* on page 9-531
0x002C	LBUFLEVEL	RO	0x--------	32	*Latched Buffer Fill Level, LBUFLEVEL* on page 9-532
0x0030	CBUFLEVEL	RO	0x--------	32	*Current Buffer Fill Level, CBUFLEVEL* on page 9-533
0x0034	BUFWM	RW	0x--------	32	*Buffer Level Water Mark, BUFWM* on page 9-534
0x0038	RRPHI	RW	0x000-----	32	*RAM Read Pointer High register, RRPHI* on page 9-535
0x003C	RWPHI	RW	0x000-----	32	*RAM Write Pointer High register, RWPHI* on page 9-536
0x0110	AXICTL	RW	0x00000-0-	32	*AXI Control Register, AXICTL* on page 9-537
0x0118	DBALO	RW	0x--------	32	*Data Buffer Address Low register, DBALO* on page 9-539
0x011C	DBAHI	RW	0x000-----	32	*Data Buffer Address HIGH register, DBAHI* on page 9-540
0x0120	RURP	WO	0x00000000	32	*RAM Update Read Pointer register, RURP* on page 9-541
0x0300	FFSR	RO	0x0000000-	32	*Formatter and Flush Status Register, FFSR* on page 9-542
0x0304	FFCR	RW	0x00000000	32	*Formatter and Flush Control Register, FFCR* on page 9-543
0x0308	PSCR	RW	0x0000000a	32	*Periodic Synchronization Counter Register, PSCR* on page 9-546
0x0EDC	ITATBMCTR0	WO	0x00000000	32	*Integration Test ATB Master Control 0 register, ITATBMCTR0* on page 9-547

Table 9-377 css600_tmc_etr - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0EE0	ITEVTINTR	WO	0x00000000	32	*Integration Test Event and Interrupt Control Register, ITEVTINTR* on page 9-548
0x0EE8	ITTRFLIN	RO	0x00000000	32	*Integration Test Trigger In and Flush In register, ITTRFLIN* on page 9-549
0x0EEC	ITATBDATA0	RO	0x00000000	32	*Integration Test ATB Data 0 Register, ITATBDATA0* on page 9-550
0x0EF0	ITATBCTR2	WO	0x00000000	32	*Integration Test ATB Control 2 Register, ITATBCTR2* on page 9-552
0x0EF4	ITATBCTR1	RO	0x00000000	32	*Integration Test ATB Control 1 Register, ITATBCTR1* on page 9-553
0x0EF8	ITATBCTR0	RO	0x00000000	32	*Integration Test ATB Control 0 Register, ITATBCTR0* on page 9-554
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-555
0x0FA0	CLAIMSET	RW	0x0000000f	32	*Claim Tag Set Register, CLAIMSET* on page 9-556
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-557
0x0FB8	AUTHSTATUS	RO	0x000000--	32	*Authentication Status Register, AUTHSTATUS* on page 9-558
0x0FC8	DEVID	RO	0x03----40	32	*Device Configuration Register, DEVID* on page 9-560
0x0FC4	DEVID1	RO	0x00000001	32	*Device Configuration Register, DEVID1* on page 9-562
0x0FCC	DEVTYPE	RO	0x00000021	32	*Device Type Identifier Register, DEVTYPE* on page 9-563
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-564
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-565
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-566
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-567
0x0FE0	PIDR0	RO	0x000000e8	32	*Peripheral Identification Register 0, PIDR0* on page 9-568
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-569
0x0FE8	PIDR2	RO	0x00000001	32	*Peripheral Identification Register 2, PIDR2* on page 9-570
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-571
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-572
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-573
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-574
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-575

9.14.2 Register descriptions

This section describes the `css600_tmc_etr` registers.

9.14.1 Register summary on page 9-520 provides cross references to individual registers.

RAM Size register, RSZ

Defines the size of trace memory in units of 32-bit words.

The RSZ register characteristics are:

Attributes

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

The following figure shows the bit assignments.

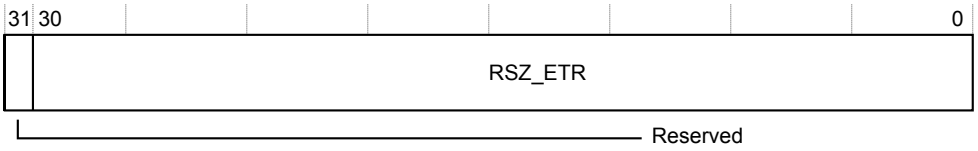

Figure 9-365 RSZ register bit assignments

The following table shows the bit assignments.

Table 9-378 RSZ register bit assignments

Bits	Reset value	Name	Function
[30:0]	UNKNOWN	RSZ_ETR	RAM size. Indicates the size of trace memory in 32-bit words. Returns the programmed value on reading.

Status register, STS

Indicates the status of the Trace Memory Controller. After a reset, software must ignore all the fields of this register except STS.TMCReady. The other fields have meaning only when the TMC has left the Disabled state. Writes to all RO fields of this register are ignored.

The STS register characteristics are:

Attributes

Offset	`0x000c`
Type	Read-write
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

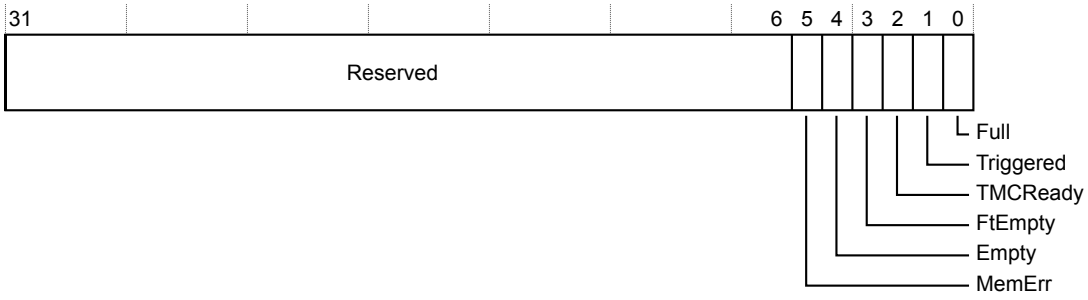

Figure 9-366 STS register bit assignments

The following table shows the bit assignments.

Table 9-379 STS register bit assignments

Bits	Reset value	Name	Function
[5]	UNKNOWN	MemErr	Memory error status. This bit indicates whether an error has occurred on the AXI master interface. The error could be due to an error response received from the connected AXI slave or due to attempted AXI transfers without proper authentication.
[4]	UNKNOWN	Empty	Trace buffer empty. If set, this bit indicates that the trace memory does not contain any valid trace data. However, this does not mean that the pipeline stages within the TMC are empty. To determine whether the internal pipeline stages are empty, the software must read the STS.TMCReady bit.
[3]	UNKNOWN	FtEmpty	Formatter pipeline empty. This bit is deprecated and is present in this register to support backwards compatibility with earlier versions of the ETB. It is set when trace capture has stopped, and all internal pipelines and buffers have drained. Unlike STS.TMCReady, it is not affected by buffer drains and AXI accesses. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.
[2]	`0b1`	TMCReady	TMC ready. This flag is set when trace capture has stopped and all internal pipelines and buffers have drained, the TMC is not draining as a result of FFCR.DrainBuffer bit being set (ETF only), and the AXI interface is not busy and the response for final AXI write has been received (ETR only) are all true. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.

Table 9-379 STS register bit assignments (continued)

Bits	Reset value	Name	Function
[1]	UNKNOWN	Triggered	TMC triggered. This bit is set when trace capture is in progress and the TMC has detected a trigger event. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. A trigger event is when the TMC has written a set number of data words, as programmed in the TRG register, into the trace memory after a rising edge of **trigin** input, or a trigger packet (**atid_s** = `0x7D`) is received in the input trace.
[0]	UNKNOWN	Full	Trace memory full. This bit helps in determining the amount of valid data present in the trace memory. Writes to this bit are allowed in Disabled state. However, it is not affected by the reprogramming of pointer registers in this state.

RAM Read Data register, RRD

Reading this register allows data to be read from the trace memory. When the memory width is greater than the APB width (32-bit), multiple reads to the same location must be performed to get the complete memory dataword. The RRP register is incremented only after the last of the required number of read accesses to fetch one memory dataword completes. When the trace memory is empty, this register returns `0xFFFFFFFF`. This value is chosen because it cannot be generated as part of trace data when the formatter is enabled.

The RRD register characteristics are:

Attributes

Offset `0x0010`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

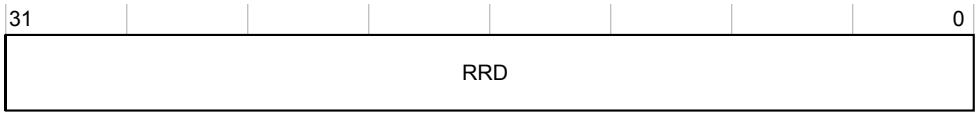

Figure 9-367 RRD register bit assignments

The following table shows the bit assignments.

Table 9-380 RRD register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RRD	Returns the data read from trace memory.

RAM Read Pointer register, RRP

The RAM Read Pointer Register contains the value of the read pointer that is used to read entries from trace memory over the APB interface. Software must program it before enabling trace capture.

The RRP register characteristics are:

Attributes

Offset `0x0014`

Type Read-write

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

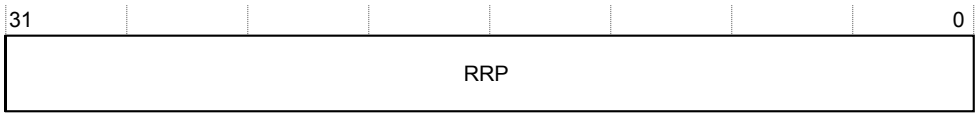

Figure 9-368 RRP register bit assignments

The following table shows the bit assignments.

Table 9-381 RRP register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RRP	When the `AXI_ADDR_WIDTH`is 32 bits or more, the RRP register is 32 bits wide and the contents of this register represents the lower 32 bits of the AXI address that is used to access trace memory on the next RRD read. When the `AXI_ADDR_WIDTH`is 0, this register is reserved and access type is RAZ/WI.

RAM Write Pointer register, RWP

RAM Write Pointer Register sets the write pointer that is used to write entries into the trace memory. Software must program it before enabling trace capture.

The RWP register characteristics are:

Attributes

Offset `0x0018`

Type Read-write

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

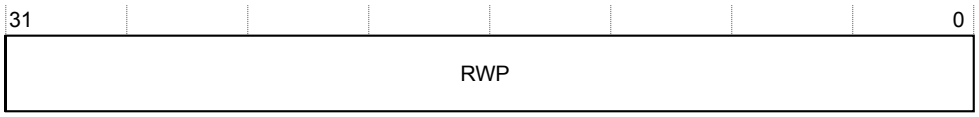

Figure 9-369 RWP register bit assignments

The following table shows the bit assignments.

Table 9-382 RWP register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RWP	When the `AXI_ADDR_WIDTH`is 32 bits or more, the RWP register is 32 bits wide and the contents of this register represents the lower 32 bits of the AXI address that is used to access trace memory on the next RWD write. When the `AXI_ADDR_WIDTH`is 0, this register is reserved and access type is RAZ/WI.

Trigger Counter register, TRG

In Circular Buffer mode, the Trigger Counter register specifies the number of 32-bit words to capture in the trace memory, after detection of either a rising edge on the **trigin** input or a trigger packet in the incoming trace stream, that is, where **atid_s** = `0x7D`. The value programmed must be aligned to the frame length of 128 bits.

The TRG register characteristics are:

Attributes

Offset `0x001c`
Type Read-write
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

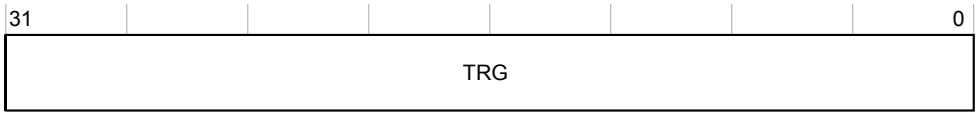

Figure 9-370 TRG register bit assignments

The following table shows the bit assignments.

Table 9-383 TRG register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	TRG	Trigger count. This count represents the number of 32-bit words of trace that are captured between a trigger packet and a trigger event. On writes, the value written must be aligned to the frame length of 128 bits.

Control Register, CTL

This register controls trace stream capture. Setting the CTL.TraceCaptEn bit to 1 enables the TMC to capture the trace data. When trace capture is enabled, the formatter behavior is controlled by the FFCR register.

The CTL register characteristics are:

Attributes

Offset `0x0020`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

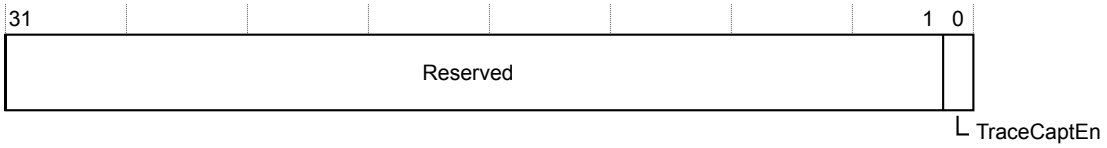

Figure 9-371 CTL register bit assignments

The following table shows the bit assignments.

Table 9-384 CTL register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	TraceCaptEn	Trace capture enable. **0** Disable trace capture. **1** Enable trace capture.

RAM Write Data register, RWD

The RAM Write Data register enables testing of trace memory connectivity to the TMC. When in Disabled state, a write to this register stores data at the location pointed to by the RWP/RWPHI registers. When not in Disabled state, writes are ignored. When the memory data width, as indicated by the DEVID.MEM_WIDTH register field, is greater than 32 bits, multiple writes to this register must be performed together to write a full memory width of data. For example, if the memory width is 128 bits, then writes to this register must be performed 4 at a time. When a full memory width of data has been written to this register, the data is written to the trace memory and the RWP is incremented to the next memory word. When the STS.MemErr bit is set, writing to this register returns an error response on the APB interface and the write data is discarded.

The RWD register characteristics are:

Attributes

Offset `0x0024`

Type Write-only

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

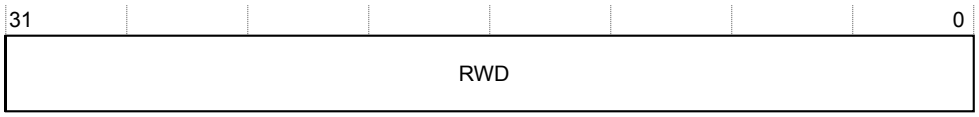

Figure 9-372 RWD register bit assignments

The following table shows the bit assignments.

Table 9-385 RWD register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	RWD	Data written to this register is placed in the trace memory.

Mode register, MODE

This register controls the TMC operating mode. The operating mode can only be changed when the TMC is in Disabled state, that is, when CTL.TraceCaptEn = 0. Attempting to write to this register in any other state results in UNPREDICTABLE behavior. The operating mode is ignored when in Disabled state.

The MODE register characteristics are:

Attributes

Offset `0x0028`
Type Read-write
Reset `0x0000000-`
Width 32

The following figure shows the bit assignments.

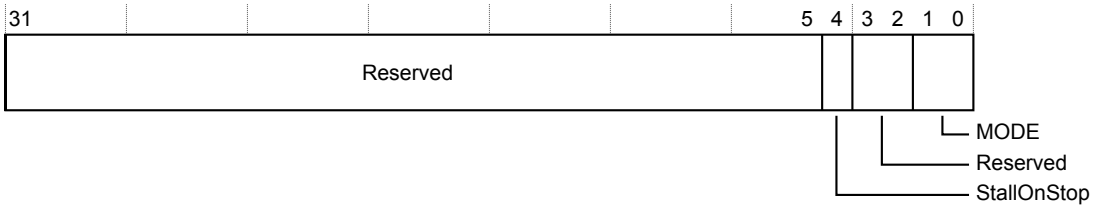

Figure 9-373 MODE register bit assignments

The following table shows the bit assignments.

Table 9-386 MODE register bit assignments

Bits	Reset value	Name	Function
[4]	`0b0`	StallOnStop	Stall On Stop. If this bit is set and the formatter stops as a result of a stop event, the output **atready_s** is de-asserted to stall the ATB interface and avoid loss of trace. If this bit is clear and the formatter stops as a result of a stop event, signal **atready_s** remains asserted but the TMC discards further incoming trace.
[1:0]	UNKNOWN	MODE	Selects the operating mode. If a reserved MODE value is programmed and trace capture is enabled, the TMC starts to operate in SWF1 mode. However, reading the MODE.MODE field returns the programmed value. `0x0` CB, Circular Buffer mode. `0x1` SWF1, Software Read FIFO mode 1. `0x2` Reserved. `0x3` SWF2, Software Read FIFO mode 2.

Latched Buffer Fill Level, LBUFLEVEL

Reading this register returns the maximum fill level of the trace memory in 32-bit words since this register was last read. Reading this register also results in its contents being updated to the current fill level. When entering Disabled state, it retains its last value. While in Disabled state, reads from this register do not affect its value. When exiting Disabled state, the LBUFLEVEL register is updated to the current fill level.

The LBUFLEVEL register characteristics are:

Attributes

Offset `0x002c`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

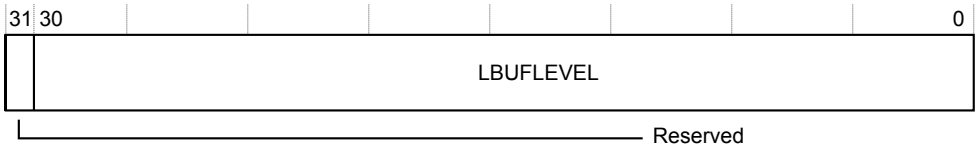

Figure 9-374 LBUFLEVEL register bit assignments

The following table shows the bit assignments.

Table 9-387 LBUFLEVEL register bit assignments

Bits	Reset value	Name	Function
[30:0]	UNKNOWN	LBUFLEVEL	Latched Buffer Fill Level. Indicates the maximum fill level of the trace memory in 32-bit words since this register was last read.

Current Buffer Fill Level, CBUFLEVEL

The CBUFLEVEL register indicates the current fill level of the trace memory in units of 32-bit words. When the TMC leaves Disabled state, this register dynamically indicates the current fill level of trace memory. It retains its value on entering Disabled state. It is not affected by the reprogramming of pointer registers in Disabled state. However, RRD reads when using the Circular Buffer in Disabled state are a special case. In this case the register indicates the updated fill level on RRD reads.

The CBUFLEVEL register characteristics are:

Attributes

Offset `0x0030`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

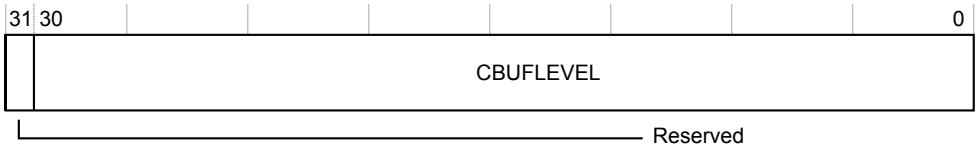

Figure 9-375 CBUFLEVEL register bit assignments

The following table shows the bit assignments.

Table 9-388 CBUFLEVEL register bit assignments

Bits	Reset value	Name	Function
[30:0]	UNKNOWN	CBUFLEVEL	Current Buffer Fill Level. Indicates the current fill level of the trace memory in 32-bit words.

Buffer Level Water Mark, BUFWM

The value that is programmed into this register indicates the desired threshold vacancy level in 32-bit words in the trace memory. When the available space in the FIFO is less than or equal to this value, that is, fill level >= (`MEM_SIZE`- BUFWM), the **full** output is asserted and the STS.Full bit is set. This register is used only in the FIFO modes, that is, SWF1, SWF2, and HWF modes. In CB mode, the same functionality is obtained by programming the RWP to the desired vacancy trigger level, so that when the pointer wraps around, the **full** output gets asserted indicating that the vacancy level has fallen below the desired level. Reading this register returns the programmed value. The maximum value that can be written into this register is `MEM_SIZE`- 1, in which case the **full** output is asserted after the first 32-bit word is written to trace memory. Writing to this register other than when in Disabled state results in UNPREDICTABLE behavior. Any software using it must program it with an initial value before setting the CTL.TraceCaptEn bit to 1.

The BUFWM register characteristics are:

Attributes

Offset	`0x0034`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

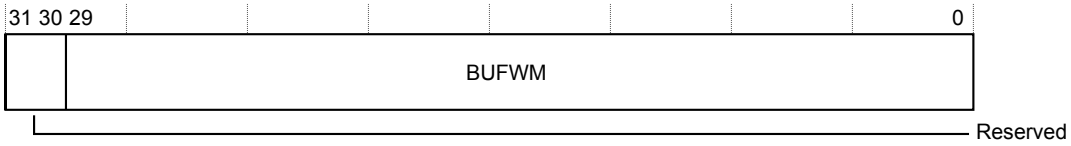

Figure 9-376 BUFWM register bit assignments

The following table shows the bit assignments.

Table 9-389 BUFWM register bit assignments

Bits	Reset value	Name	Function
[29:0]	UNKNOWN	BUFWM	Buffer Level Watermark. Indicates the desired threshold vacancy level in 32-bit words in the trace memory.

RAM Read Pointer High register, RRPHI

The RAM Read Pointer High register contains address bits >= bit[32] of the read pointer that is used to read entries from trace memory over the APB interface.

The RRPHI register characteristics are:

Attributes

Offset `0x0038`
Type Read-write
Reset `0x000-----`
Width 32

The following figure shows the bit assignments.

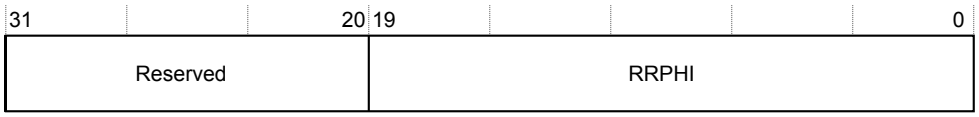

Figure 9-377 RRPHI register bit assignments

The following table shows the bit assignments.

Table 9-390 RRPHI register bit assignments

Bits	Reset value	Name	Function
[19:0]	UNKNOWN	RRPHI	RAM Read Pointer High. Bits [32] and above of the RAM read pointer.

RAM Write Pointer High register, RWPHI

The RAM Write Pointer High register sets bits >= bit[32] of the write pointer that is used to write entries into the trace memory. Software must program it before enabling trace capture.

The RWPHI register characteristics are:

Attributes

Offset `0x003c`
Type Read-write
Reset `0x000-----`
Width 32

The following figure shows the bit assignments.

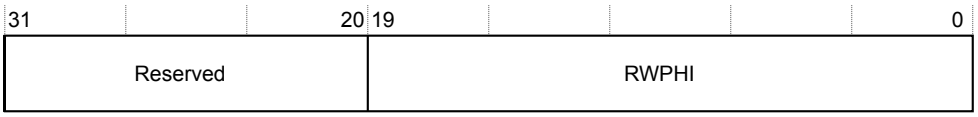

Figure 9-378 RWPHI register bit assignments

The following table shows the bit assignments.

Table 9-391 RWPHI register bit assignments

Bits	Reset value	Name	Function
[19:0]	UNKNOWN	RWPHI	RAM Write Pointer High. Bits [32] and above of the RAM write pointer.

AXI Control Register, AXICTL

This register controls TMC accesses to system memory through the AXI interface. The TMC only performs data accesses, so the **arprot_m[2]** and **awprot_m[2]** outputs are LOW for all AXI transfers.

The AXICTL register characteristics are:

Attributes

Offset `0x0110`
Type Read-write
Reset `0x00000-0-`
Width 32

The following figure shows the bit assignments.

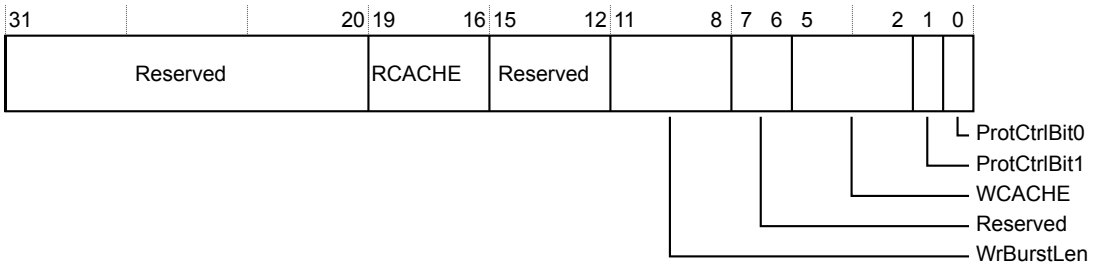

Figure 9-379 AXICTL register bit assignments

The following table shows the bit assignments.

Table 9-392 AXICTL register bit assignments

Bits	Reset value	Name	Function
[19:16]	`0b0000`	RCACHE	This field controls the AXI cache encoding for read transfers, that is, the value to be driven on the **arcache_m[3:0]** bus. Software must only program a valid AXI3 or AXI4 cache encoding value in this field. If the software attempts to program an invalid value, `0x0` is written to this field instead. Valid encodings are: **Bit[0]** **0** Non-bufferable **1** Bufferable **Bit[1]** **0** Non-cacheable **1** Cacheable **Bit[2]** **0** No Read-allocate **1** Read-allocate **Bit[3]** **0** No Write-allocate **1** Write-allocate
[11:8]	UNKNOWN	WrBurstLen	Write Burst Length. This field indicates the maximum number of data transfers that can occur within each burst that is initiated by the TMC on the AXI interface. The write burst that is initiated on the AXI can be shorter than the programmed value in a case when the formatter has stopped due to a stop condition having occurred.

Table 9-392 AXICTL register bit assignments (continued)

Bits	Reset value	Name	Function
[5:2]	`0b0000`	WCACHE	This field controls the AXI cache encoding for write transfers, that is, the value to be driven on the **awcache_m[3:0]** bus. Software must only program a valid AXI3 or AXI4 cache encoding value in this field. If software attempts to program an invalid value, `0x0` is written to this field instead. Valid encodings are: **Bit[0]** **0** Non-bufferable **1** Bufferable **Bit[1]** **0** Non-cacheable **1** Cacheable **Bit[2]** **0** No Read-allocate **1** Read-allocate **Bit[3]** **0** No Write-allocate **1** Write-allocate
[1]	UNKNOWN	ProtCtrlBit1	Secure Access (AXI4 definition). This bit controls the value that is driven on **arprot_m[1]** or **awprot_m[1]** on the AXI master interface when performing AXI transfers.
[0]	UNKNOWN	ProtCtrlBit0	Privileged Access (AXI4 definition). This bit controls the value that is driven on **arprot_m[0]** or **awprot_m[0]** on the AXI master interface when performing AXI transfers.

Data Buffer Address Low register, DBALO

This register, together with the DBAHI register, enables the TMC to locate the trace buffer in system memory. This register contains bits [31:0] of the start address of the trace buffer in system memory. This register is 32-bits wide if `AXI_ADDR_WIDTH`is >= 32-bits. If the `AXI_ADDR_WIDTH`is 0, this register is reserved and access type is RAZ/WI. Software must program it before enabling trace capture, and the programmed value must be aligned to the Trace Memory Data Width and the Frame Width. Programming an unaligned value results in UNPREDICTABLE behavior. Modifying this register other than when in Disabled state results in UNPREDICTABLE behavior.

The DBALO register characteristics are:

Attributes

Offset	`0x0118`
Type	Read-write
Reset	`0x--------`
Width	32

The following figure shows the bit assignments.

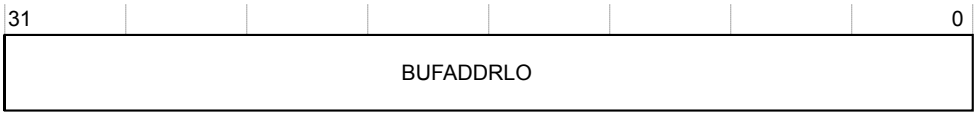

Figure 9-380 DBALO register bit assignments

The following table shows the bit assignments.

Table 9-393 DBALO register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	BUFADDRLO	Data Buffer Low Address. Holds the lower 32 bits of the AXI address that is used to locate the trace buffer in system memory.

Data Buffer Address HIGH register, DBAHI

This register, together with the DBALO register, enables the TMC to locate the trace buffer in system memory. It contains bits >= bit[32] of the start address of the trace buffer in system memory. The width of this register is given by: (`AXI_ADDR_WIDTH`- 32), however, if `AXI_ADDR_WIDTH`is <= 32 bits, this register is reserved and access type is RAZ/WI. Modifying this register, other than when in Disabled state, results in UNPREDICTABLE behavior. Software must program it with an initial value before setting CTL.TraceCaptEn bit to 1.

The DBAHI register characteristics are:

Attributes

Offset `0x011c`

Type Read-write

Reset `0x000-----`

Width 32

The following figure shows the bit assignments.

Figure 9-381 DBAHI register bit assignments

The following table shows the bit assignments.

Table 9-394 DBAHI register bit assignments

Bits	Reset value	Name	Function
[19:0]	UNKNOWN	BUFADDRHI	Data Buffer High Address. Holds the upper bits, that is, bit[32] and above, of the AXI address that is used to locate the trace buffer in the system.

RAM Update Read Pointer register, RURP

The RURP register enables software to inform the TMC of the amount of trace data that is extracted directly from system memory in SWF2 mode. Writes to this register are ignored when the TMC is in Disabled state or when not in SWF2 mode.

The RURP register characteristics are:

Attributes

Offset `0x0120`
Type Write-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

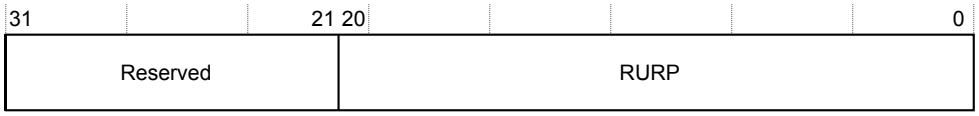

Figure 9-382 RURP register bit assignments

The following table shows the bit assignments.

Table 9-395 RURP register bit assignments

Bits	Reset value	Name	Function
[20:0]	`0x0`	RURP	RAM Update Read Pointer. A write to the RURP register causes the TMC to update the RAM Read Pointer, both the RRP and RRPHI registers, based on the value that is written to it. RURP allows up to 1MB of data to be extracted in a single chunk. Reads always return `0x0`. The following constraints apply to the write values: `0x000000` - no effect, `0x000001-0x100000` - increment RRP by this value, `0x100001-0x1FFFFF` - reserved.

Formatter and Flush Status Register, FFSR

This register indicates the status of the Formatter, and the status of Flush request.

The FFSR register characteristics are:

Attributes

Offset	`0x0300`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

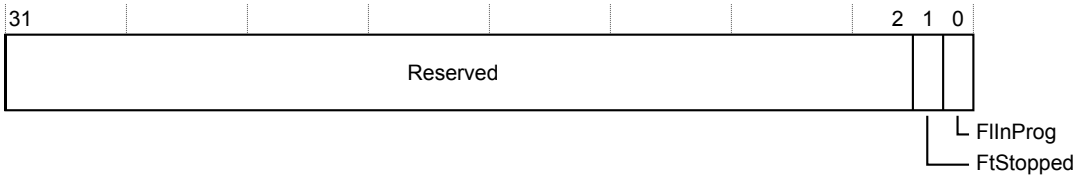

Figure 9-383 FFSR register bit assignments

The following table shows the bit assignments.

Table 9-396 FFSR register bit assignments

Bits	Reset value	Name	Function
[1]	UNKNOWN	FtStopped	Formatter Stopped. This bit behaves the same way as STS.FtEmpty. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. The FFCR.FtStopped bit is deprecated and is present in this register to support backwards-compatibility with earlier versions of the ETB. **0** Trace capture has not yet completed. **1** Trace capture has completed and all captured trace data has been written to the trace memory.
[0]	UNKNOWN	FlInProg	Flush In Progress. This bit indicates whether the TMC is currently processing a flush request. In the ETB and ETR configurations, the flush initiation is controlled by the flush control bits in the FFCR register. In the ETF configuration, the flush request can also be from the ATB Master port. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. When in Disabled state, this bit is not updated. **0** No flush activity in progress. **1** Flush in progress on the ATB slave interface or the TMC internal pipeline.

Formatter and Flush Control Register, FFCR

This register allows user control of the stop, trigger, and flush events.

The FFCR register characteristics are:

Attributes

Offset	`0x0304`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

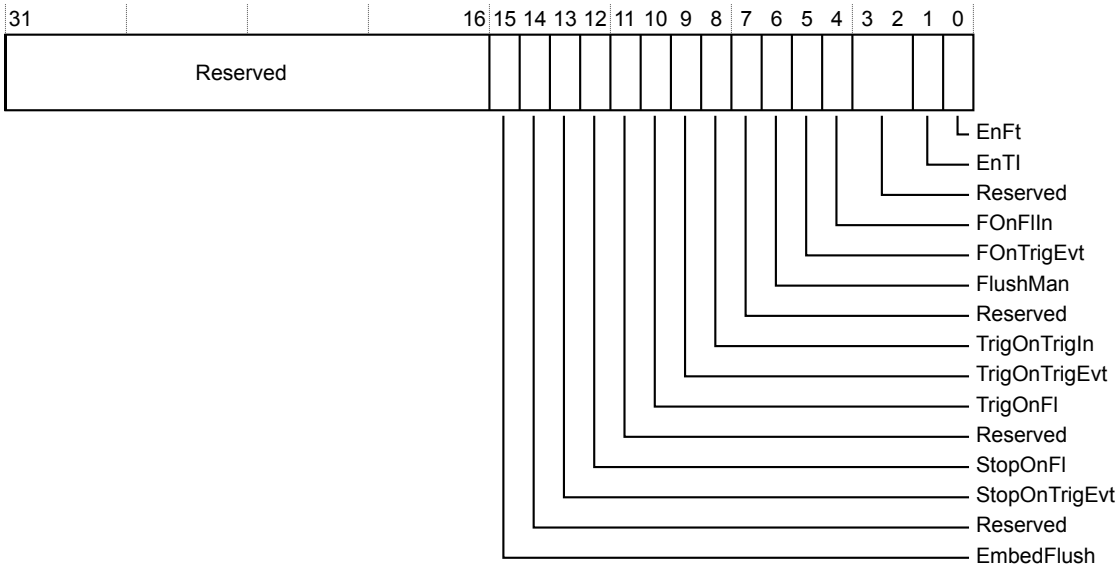

Figure 9-384 FFCR register bit assignments

The following table shows the bit assignments.

Table 9-397 FFCR register bit assignments

Bits	Reset value	Name	Function
[15]	`0b0`	EmbedFlush	Embed Flush ID (flush completion packet). Enables insertion of Flush ID `0x7B` with a single byte of data payload = `0x00` in the output trace, immediately after the last flush data byte, when a flush completes on the ATB slave interface. This bit is effective only in Normal formatting modes. In Bypass mode, the Flush ID insertion remains disabled and this bit is ignored. **0** Disable Flush ID insertion. **1** Enable Flush ID insertion.
[13]	`0b0`	StopOnTrigEvt	Stop On Trigger Event. If this bit is set, the formatter is stopped when a Trigger Event has been observed. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.

Table 9-397 FFCR register bit assignments (continued)

Bits	Reset value	Name	Function
[12]	0b0	StopOnFl	Stop On Flush. If this bit is set, the formatter is stopped on completion of a flush operation. The initiation of a flush operation is controlled by programming the register bits FFCR.FlushMan, FFCR.FOnTrigEvt, and FFCR.FOnFlIn. When a flush-initiation condition occurs, **afvalid_s** is asserted, and when the flush completion is received, that is, **afready_s**=1, trace capture is stopped. Any remaining data in the formatter is appended with a post-amble and written to trace memory. The flush operation is then complete. When the TMC is configured as an ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a formatter stop regardless of the value that is programmed in this bit.
[10]	0b0	TrigOnFl	Indicate on trace stream the completion of flush. If this bit is set, a trigger is indicated on the trace stream when **afready_s** is received for a flush in progress. If this bit is clear, no triggers are embedded in the trace stream on flush completion. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. When the TMC is configured as ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a trigger indication on the trace stream regardless of the value that is programmed in this bit.
[9]	0b0	TrigOnTrigEvt	Indicate on trace stream the occurrence of a Trigger Event. If this bit is set, a trigger is indicated on the output trace stream when a Trigger Event occurs. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[8]	0b0	TrigOnTrigIn	Indicate on trace stream the occurrence of a rising edge on **trigin**. If this bit is set, a trigger is indicated on the trace stream when a rising edge is detected on the **trigin** input. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit.
[6]	0b0	FlushMan	Manually generate a flush of the system. Writing 1 to this bit causes a flush to be generated. This bit is cleared automatically when, in formatter bypass mode, **afready_s** was sampled high, or, in normal formatting mode, **afready_s** was sampled high and all flush data was output to the trace memory. If CTL.TraceCaptEn=0, writes to this bit are ignored.
[5]	0b0	FOnTrigEvt	Flush on Trigger Event. Setting this bit generates a flush when a Trigger Event occurs. If FFCR.StopOnTrigEvt is set, this bit is ignored. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[4]	0b0	FOnFlIn	Setting this bit enables the detection of transitions on the **flushin** input by the TMC. If this bit is set and the formatter has not already stopped, a rising edge on **flushin** initiates a flush request.

Table 9-397 FFCR register bit assignments (continued)

Bits	Reset value	Name	Function
[1]	0b0	EnTI	Enable Trigger Insertion. Setting this bit enables the insertion of triggers in the formatted trace stream. A trigger is indicated by inserting one byte of data 0x00 with **atid_s**=0x7D in the trace stream. Trigger indication on the trace stream is also controlled by the register bits FFCR.TrigOnFl, FFCR.TrigOnTrigEvt, and FFCR.TrigOnTrigIn. This bit can only be changed when the TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled.
[0]	0b0	EnFt	Enable Formatter. If this bit is set, formatting is enabled. This bit takes effect when not in Disabled state, and is ignored in Disabled state. If this bit is clear, formatting is disabled. In this case, the incoming trace data is assumed to be from a single trace source. If multiple trace IDs are received by the TMC when trace capture is enabled and the formatter is disabled, it results in interleaving of trace data. Disabling of formatting is deprecated, and is supported in the TMC for backwards-compatibility with earlier versions of the ETB. Therefore, disabling of formatting is supported only in CB mode. Features in the TMC such as the FIFO modes and the FFCR.DrainBuffer bit that are not part of the earlier versions of the ETB do not support disabling of formatting. This bit can only be changed when TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled. If the TMC is enabled in a mode other than Circular Buffer mode with formatting disabled, it results in formatting being enabled.

Periodic Synchronization Counter Register, PSCR

This register determines the reload value of the Periodic Synchronization Counter. This counter enables the frequency of sync packets to be optimized to the trace capture buffer size. The default behavior of the counter is to generate periodic synchronization requests, **syncreq_s**, on the ATB slave interface.

The PSCR register characteristics are:

Attributes

Offset	`0x0308`
Type	Read-write
Reset	`0x0000000a`
Width	32

The following figure shows the bit assignments.

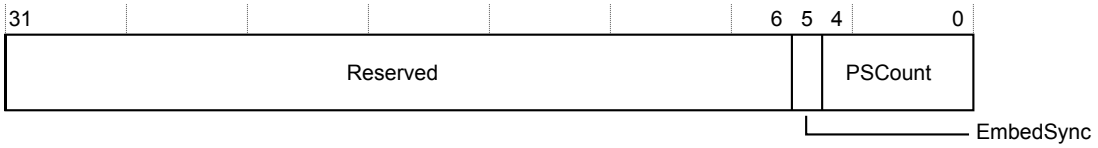

Figure 9-385 PSCR register bit assignments

The following table shows the bit assignments.

Table 9-398 PSCR register bit assignments

Bits	Reset value	Name	Function
[5]	`0b0`	EmbedSync	Embed Frame Sync Packet in the trace stream. Setting this bit to 1 enables the formatter to insert frame sync packets in the trace stream at periodic intervals. If this bit is set and the Synchronization Counter is enabled, the formatter inserts a 32-bit frame sync packet in the trace stream when the counter reaches 0. This bit is effective only when formatting is enabled, that is when FFCR.EnTI=1 or FFCR.EnFt=1, and it is ignored when the formatter is in bypass mode.
[4:0]	`0b01010`	PSCount	Periodic Synchronization Count. Determines the reload value of the Synchronization Counter. The reload value takes effect the next time the counter reaches zero. When trace capture is enabled, the Synchronization Counter counts the number of bytes of trace data that is stored into the trace memory, regardless of whether the trace data has been formatted by the TMC or not, since the occurrence of the last sync request on the ATB slave interface. When the counter reaches 0, a sync request is sent on the ATB slave interface. Reads from this register return the reload value that is programmed in this register. This field resets to `0x0A`, that is, the default sync period is 2^10 bytes. If a reserved value is programmed in this register field, the value `0x1B` is used instead, and subsequent reads from this register also return `0x1B`. The following constraints apply to the values written to the PSCount field: `0x0` - synchronization is disabled, `0x1-0x6` - reserved, `0x7-0x1B` - synchronization period is 2^PSCount bytes. The smallest value `0x7` gives a sync period of 128 bytes. The maximum allowed value `0x1B` gives a sync period of 2^27 bytes, `0x1C-0x1F` - reserved.

Integration Test ATB Master Control 0 register, ITATBMCTR0

This register enables control of the ATB master outputs **atbytes_m**, **atwakeup_m**, **afready_m**, and **atvalid_m** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit, and the reads return `0x0`.

The ITATBMCTR0 register characteristics are:

Attributes

Offset	`0x0edc`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

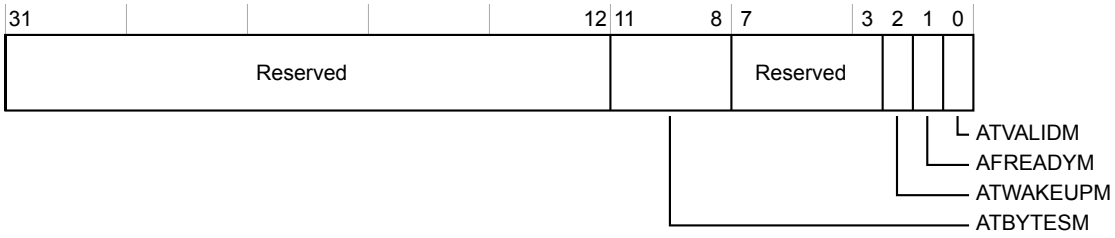

Figure 9-386 ITATBMCTR0 register bit assignments

The following table shows the bit assignments.

Table 9-399 ITATBMCTR0 register bit assignments

Bits	Reset value	Name	Function
[11:8]	`0b0000`	ATBYTESM	Controls the value of **atbytes_m** output in integration mode. This width of this field is given by N=8+$\log_2$(ATB DATA WIDTH/8).
[2]	`0b0`	ATWAKEUPM	Controls the value of **atwakeup_m** output in integration mode.
[1]	`0b0`	AFREADYM	Controls the value of **afready_m** output in integration mode.
[0]	`0b0`	ATVALIDM	Controls the value of **atvalid_m** output in integration mode.

Integration Test Event and Interrupt Control Register, ITEVTINTR

This register controls the values of event and interrupt outputs in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITEVTINTR register characteristics are:

Attributes

Offset	`0x0ee0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

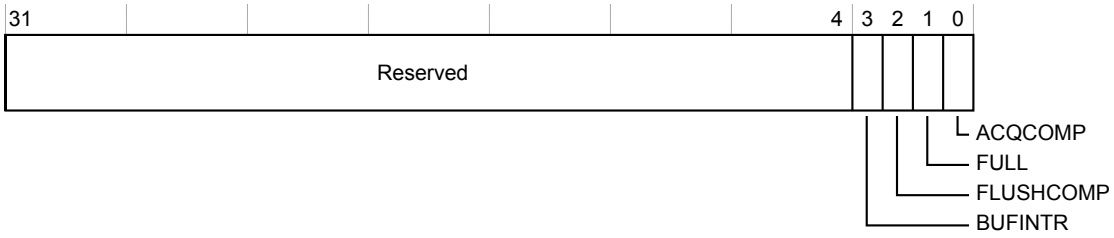

Figure 9-387 ITEVTINTR register bit assignments

The following table shows the bit assignments.

Table 9-400 ITEVTINTR register bit assignments

Bits	Reset value	Name	Function
[3]	`0b0`	BUFINTR	Controls the value of **bufintr** output in integration mode.
[2]	`0b0`	FLUSHCOMP	Controls the value of **flushcomp** output in integration mode.
[1]	`0b0`	FULL	Controls the value of **full** output in integration mode.
[0]	`0b0`	ACQCOMP	Controls the value of **acqcomp** output in integration mode.

Integration Test Trigger In and Flush In register, ITTRFLIN

This register captures the values of the **flushin** and **trigin** inputs in integration mode. In functional mode, this register behaves as RAZ/WI.

The ITTRFLIN register characteristics are:

Attributes

Offset	`0x0ee8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

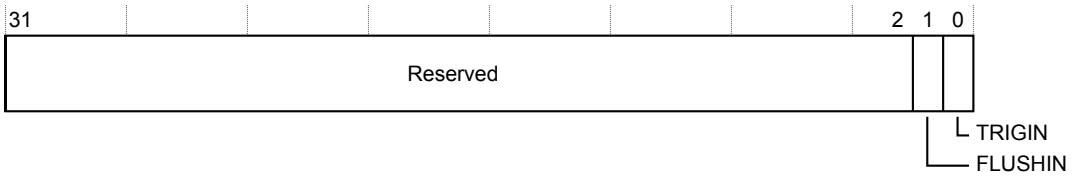

Figure 9-388 ITTRFLIN register bit assignments

The following table shows the bit assignments.

Table 9-401 ITTRFLIN register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	FLUSHIN	Integration status of **flushin** input. In integration mode, this bit latches to 1 on a rising edge of the **flushin** input. It is cleared when the register is read or when integration mode is disabled.
[0]	`0b0`	TRIGIN	Integration status of **trigin** input. In integration mode, this bit latches to 1 on a rising edge of the **trigin** input. It is cleared when the register is read or when integration mode is disabled.

Integration Test ATB Data 0 Register, ITATBDATA0

This register captures the value of **atdata_s** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding **atdata_s** bits. The width of this register is given by: 1+(ATB DATA WIDTH)/8.

The ITATBDATA0 register characteristics are:

Attributes

Offset `0x0eec`
Type Read-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

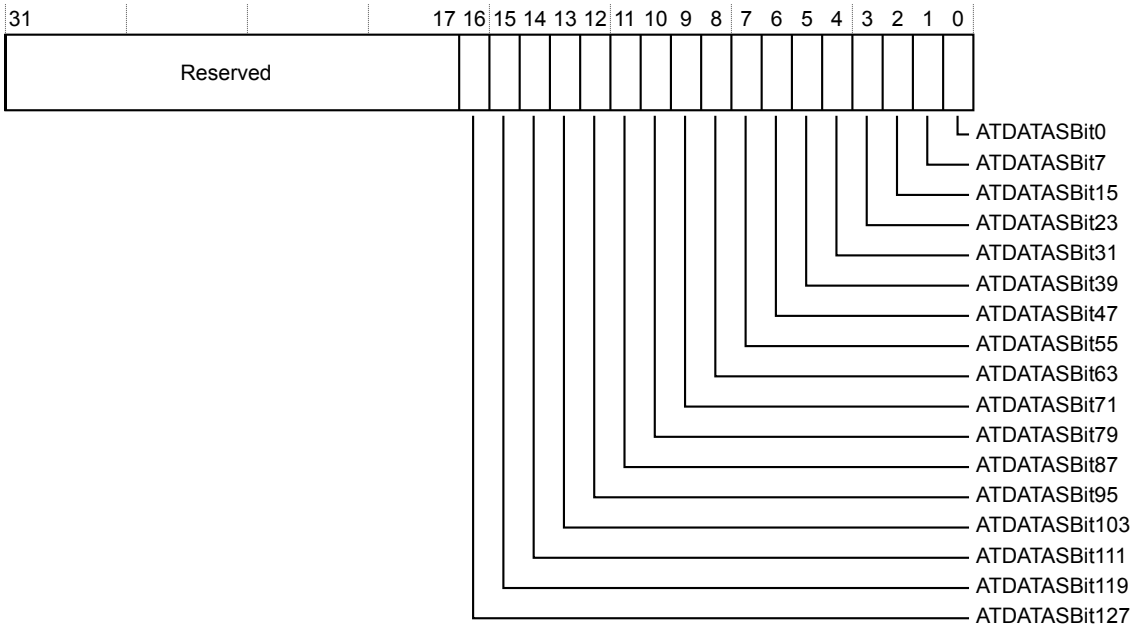

Figure 9-389 ITATBDATA0 register bit assignments

The following table shows the bit assignments.

Table 9-402 ITATBDATA0 register bit assignments

Bits	Reset value	Name	Function
[16]	`0b0`	ATDATASBit127	Returns the value of **atdata_s[127]** input in integration mode.
[15]	`0b0`	ATDATASBit119	Returns the value of **atdata_s[119]** input in integration mode.
[14]	`0b0`	ATDATASBit111	Returns the value of **atdata_s[111]** input in integration mode.
[13]	`0b0`	ATDATASBit103	Returns the value of **atdata_s[103]** input in integration mode.
[12]	`0b0`	ATDATASBit95	Returns the value of **atdata_s[95]** input in integration mode.
[11]	`0b0`	ATDATASBit87	Returns the value of **atdata_s[87]** input in integration mode.
[10]	`0b0`	ATDATASBit79	Returns the value of **atdata_s[79]** input in integration mode.
[9]	`0b0`	ATDATASBit71	Returns the value of **atdata_s[71]** input in integration mode.
[8]	`0b0`	ATDATASBit63	Returns the value of **atdata_s[63]** input in integration mode.

Table 9-402 ITATBDATA0 register bit assignments (continued)

Bits	Reset value	Name	Function
[7]	0b0	ATDATASBit55	Returns the value of **atdata_s[55]** input in integration mode.
[6]	0b0	ATDATASBit47	Returns the value of **atdata_s[47]** input in integration mode.
[5]	0b0	ATDATASBit39	Returns the value of **atdata_s[39]** input in integration mode.
[4]	0b0	ATDATASBit31	Returns the value of **atdata_s[31]** input in integration mode.
[3]	0b0	ATDATASBit23	Returns the value of **atdata_s[23]** input in integration mode.
[2]	0b0	ATDATASBit15	Returns the value of **atdata_s[15]** input in integration mode.
[1]	0b0	ATDATASBit7	Returns the value of **atdata_s[7]** input in integration mode.
[0]	0b0	ATDATASBit0	Returns the value of **atdata_s[0]** input in integration mode.

Integration Test ATB Control 2 Register, ITATBCTR2

This register enables control of ATB slave outputs **atready_s**, **afvalid_s**, and **syncreq_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITATBCTR2 register characteristics are:

Attributes

Offset	`0x0ef0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

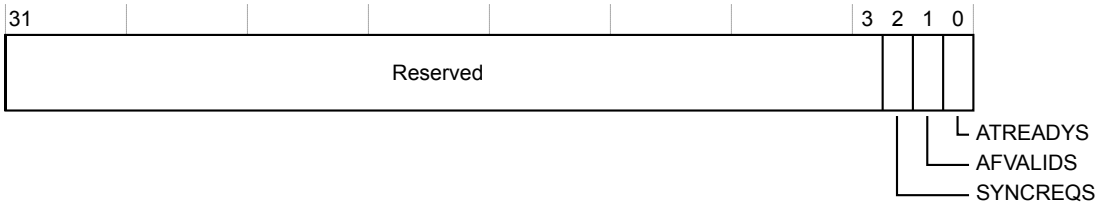

Figure 9-390 ITATBCTR2 register bit assignments

The following table shows the bit assignments.

Table 9-403 ITATBCTR2 register bit assignments

Bits	Reset value	Name	Function
[2]	`0b0`	SYNCREQS	Controls the value of **syncreq_s** output in integration mode.
[1]	`0b0`	AFVALIDS	Controls the value of **afvalid_s** output in integration mode.
[0]	`0b0`	ATREADYS	Controls the value of **atready_s** output in integration mode.

Integration Test ATB Control 1 Register, ITATBCTR1

This register captures the value of the **atid_s[6:0]** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of **atid_s** input.

The ITATBCTR1 register characteristics are:

Attributes

Offset `0x0ef4`

Type Read-only

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

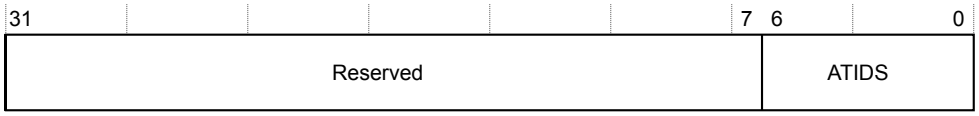

Figure 9-391 ITATBCTR1 register bit assignments

The following table shows the bit assignments.

Table 9-404 ITATBCTR1 register bit assignments

Bits	Reset value	Name	Function
[6:0]	`0b0000000`	ATIDS	Returns the value of **atid_s[6:0]** input in integration mode.

Integration Test ATB Control 0 Register, ITATBCTR0

This register captures the values of ATB slave inputs **atvalid_s**, **afready_s**, **atwakeup_s**, and **atbytes_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding input pins. The width of this register is given by: $8+\log_2(\text{ATB DATA WIDTH}/8)$.

The ITATBCTR0 register characteristics are:

Attributes

Offset	`0x0ef8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

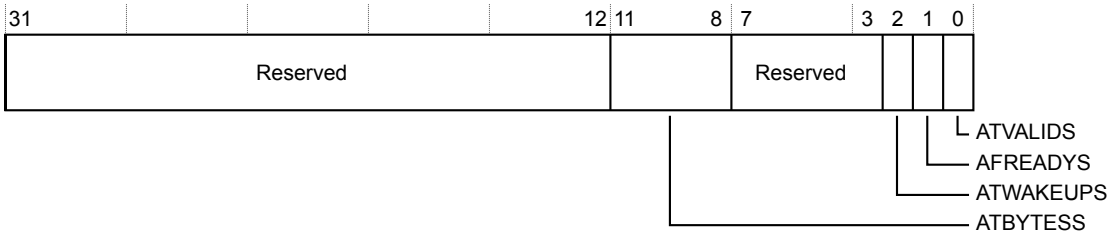

Figure 9-392 ITATBCTR0 register bit assignments

The following table shows the bit assignments.

Table 9-405 ITATBCTR0 register bit assignments

Bits	Reset value	Name	Function
[11:8]	`0b0000`	ATBYTESS	Returns the value of **atbytes_s** input in integration mode. $N=8+\log_2(\text{ATB DATA WIDTH}/8)$.
[2]	`0b0`	ATWAKEUPS	Returns the value of **atwakeup_s** input in integration mode.
[1]	`0b0`	AFREADYS	Returns the value of **afready_s** input in integration mode.
[0]	`0b0`	ATVALIDS	Returns the value of **atvalid_s** input in integration mode.

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.

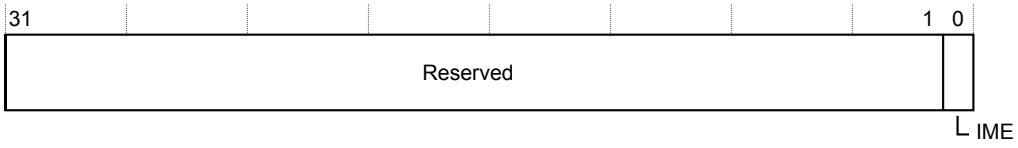

Figure 9-393 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-406 ITCTRL register bit assignments

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

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 `0x0000000f`
Width 32

The following figure shows the bit assignments.

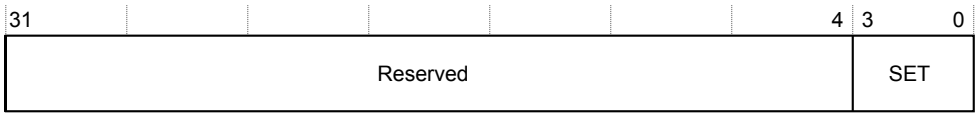

Figure 9-394 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-407 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	SET	A bit programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

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.

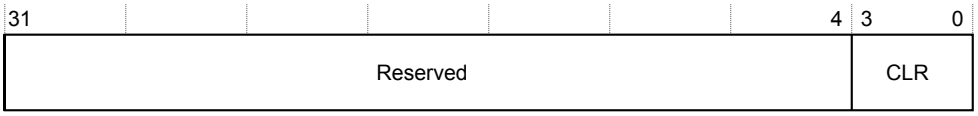

Figure 9-395 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-408 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	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.

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.

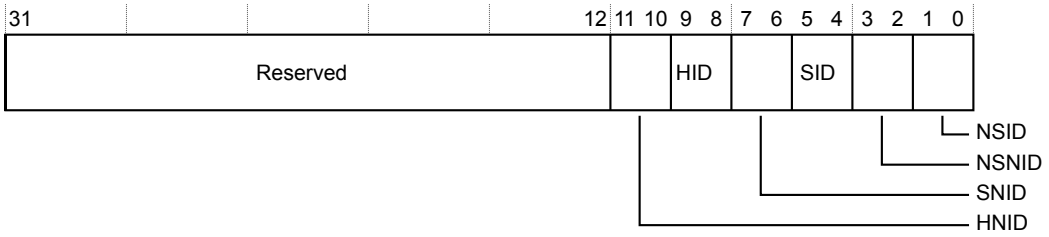

Figure 9-396 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-409 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	IMPLEMENTATION_DEFINED	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-409 AUTHSTATUS register bit assignments (continued)

Bits	Reset value	Name	Function
[3:2]	0b00	NSNID	Non-secure non-invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.
[1:0]	IMPLEMENTATION_DEFINED	NSID	Non-secure invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.

Device Configuration Register, DEVID

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

The DEVID register characteristics are:

Attributes

Offset	`0x0fc8`
Type	Read-only
Reset	`0x03----40`
Width	32

The following figure shows the bit assignments.

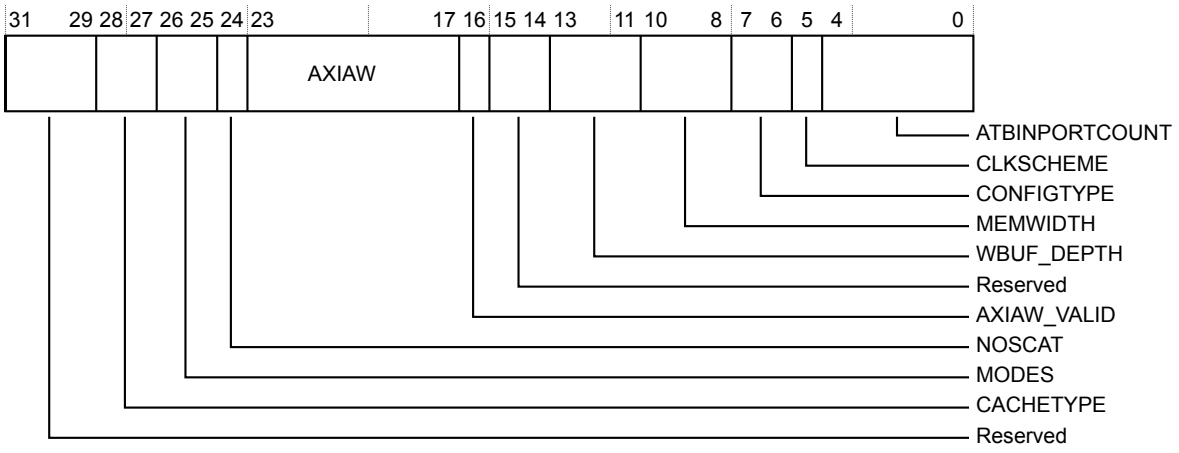

Figure 9-397 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-410 DEVID register bit assignments

Bits	Reset value	Name	Function
[28:27]	`0b00`	CACHETYPE	Indicates the format of BUSCTL register bus control fields. Reads as 2'b00 indicating that AXICTL bus attribute bits [19:16] and [5:2] follow an implementation-defined non-generic format. See AXICTL register description.
[26:25]	`0b01`	MODES	Indicates the supported modes of operation. Reads as 2'b01 indicating that ETR supports CB, SWF1, and SWF2 modes.
[24]	`0b1`	NOSCAT	Indicates whether the scatter-gather mode is implemented. Fixed at 1 indicating that scatter-gather mode is not implemented.
[23:17]	IMPLEMENTATION_DEFINED	AXIAW	This field indicates the width of AXI address bus in ETR configuration. This field is valid only when DEVID.AXIAW_VALID is set. Possible values are: `0x20` 32-bit AXI address buses. `0x28` 40-bit AXI address buses. `0x2C` 44-bit AXI address buses. `0x30` 48-bit AXI address buses. `0x34` 52-bit AXI address buses.

Table 9-410 DEVID register bit assignments (continued)

Bits	Reset value	Name	Function
[16]	`0b1`	AXIAW_VALID	Indicates whether field DEVID.AW is valid. The value of this field is fixed at 1.
[13:11]	IMPLEMENTATION_DEFINED	WBUF_DEPTH	Log_2 of the number of write buffer entries. This value is set by the parameter `WBUFFER_DEPTH`. Each entry is of size `ATB_DATA_WIDTH`. `0x2` Depth of Write buffer is 4 entries. `0x3` Depth of Write buffer is 8 entries. `0x4` Depth of Write buffer is 16 entries. `0x5` Depth of Write buffer is 32 entries. `0x6` Depth of Write buffer is 64 entries. `0x7` Depth of Write buffer is 128 entries.
[10:8]	IMPLEMENTATION_DEFINED	MEMWIDTH	This value is equal to `ATB_DATA_WIDTH`. `0x2` Memory interface databus is 32-bits wide. `0x3` Memory interface databus is 64-bits wide. `0x4` Memory interface databus is 128-bits wide. `0x5` Memory interface databus is 256-bits wide.
[7:6]	`0b01`	CONFIGTYPE	Returns `0x0` indicating ETR configuration.
[5]	`0b0`	CLKSCHEME	RAM Clocking Scheme. This value indicates the TMC RAM clocking scheme used, that is, whether the TMC RAM operates synchronously or asynchronously to the TMC clock. Fixed to 0 indicating that TMC RAM clock is synchronous to the **clk** input.
[4:0]	`0b00000`	ATBINPORTCOUNT	Hidden Level of ATB input multiplexing. This value indicates the type/number of ATB multiplexing present on the input ATB. Fixed to `0x00` indicating that no multiplexing is present.

Device Configuration Register, DEVID1

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

The DEVID1 register characteristics are:

Attributes

Offset `0x0fc4`

Type Read-only

Reset `0x00000001`

Width 32

The following figure shows the bit assignments.

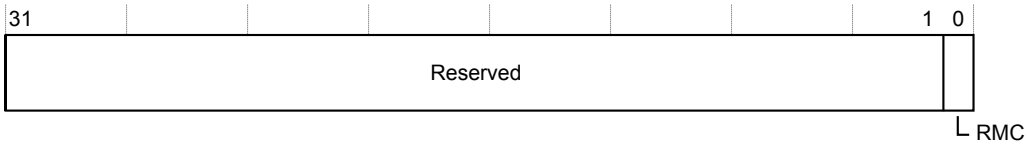

Figure 9-398 DEVID1 register bit assignments

The following table shows the bit assignments.

Table 9-411 DEVID1 register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x00000000`	-	Reserved.
[0]	1	RMC	Indicates register management mode 1 is implemented.

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 `0x00000021`
Width 32

The following figure shows the bit assignments.

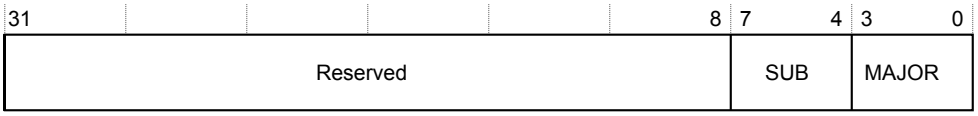

Figure 9-399 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-412 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0010`	SUB	Minor classification. Returns `0x2`, indicating this component is a Buffer.
[3:0]	`0b0001`	MAJOR	Major classification. Returns `0x1`, indicating this component is a Trace Sink.

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.

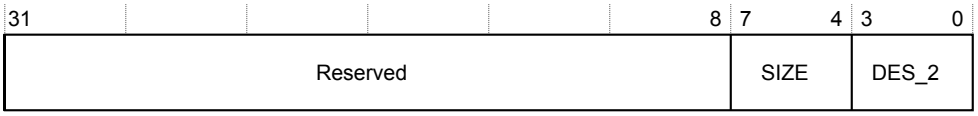

Figure 9-400 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-413 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

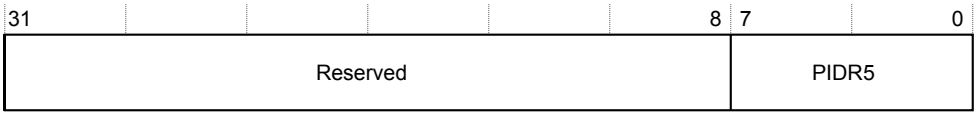

Figure 9-401 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-414 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

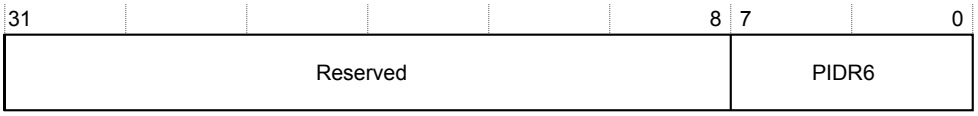

Figure 9-402 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-415 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

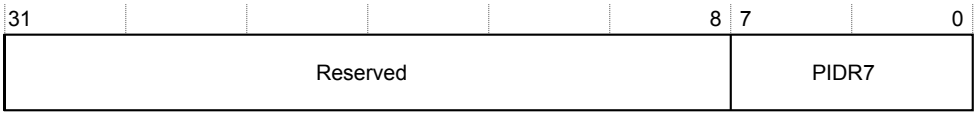

Figure 9-403 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-416 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000e8`

Width 32

The following figure shows the bit assignments.

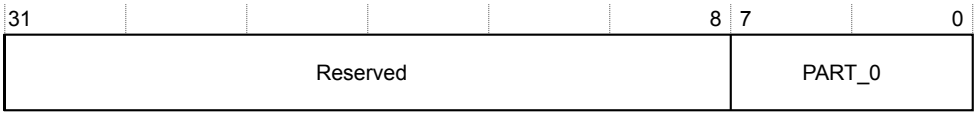

Figure 9-404 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-417 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11101000`	PART_0	Part number (lower 8 bits). Returns `0xe8`, indicating TMC ETR/ETS.

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.

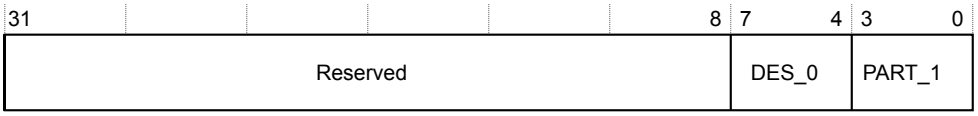

Figure 9-405 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-418 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x00000001`
Width 32

The following figure shows the bit assignments.

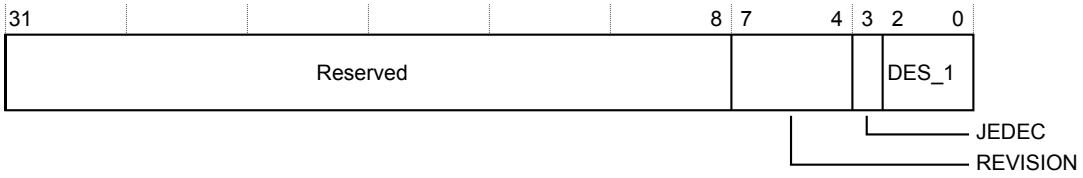

Figure 9-406 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-419 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0x1`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

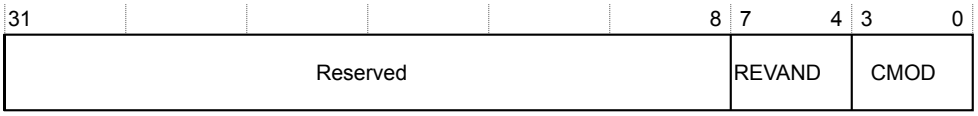

Figure 9-407 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-420 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

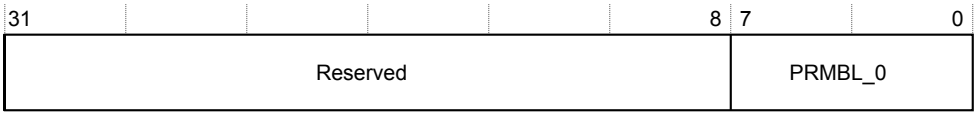

Figure 9-408 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-421 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

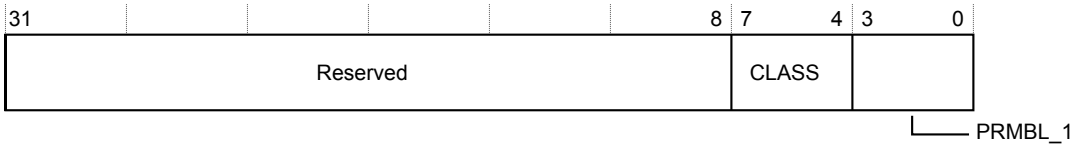

Figure 9-409 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-422 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

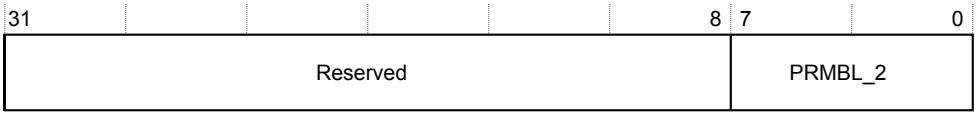

Figure 9-410 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-423 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

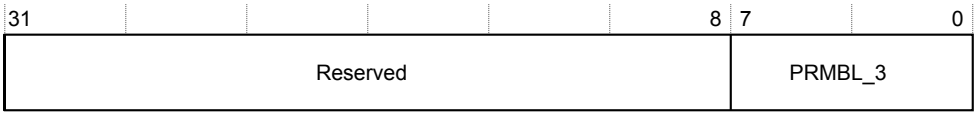

Figure 9-411 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-424 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.15 css600_tmc_ets introduction

This section describes the functions and programmers model of the `css600_tmc_ets`.

This section contains the following subsections:

- *9.15.1 Register summary* on page 9-576.
- *9.15.2 Register descriptions* on page 9-577.

9.15.1 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.

Table 9-425 css600_tmc_ets - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0004	RSZ	RW	0x0000000-	32	*RAM Size register, RSZ* on page 9-578
0x000C	STS	RW	0x000000--	32	*Status register, STS* on page 9-579
0x0010	RRD	RO	0x--------	32	*RAM Read Data register, RRD* on page 9-580
0x001C	TRG	RW	0x--------	32	*Trigger Counter register, TRG* on page 9-581
0x0020	CTL	RW	0x00000000	32	*Control Register, CTL* on page 9-582
0x0024	RWD	WO	0x00000000	32	*RAM Write Data register, RWD* on page 9-583
0x0028	MODE	RW	0x0000000-	32	*Mode register, MODE* on page 9-584
0x0300	FFSR	RO	0x0000000-	32	*Formatter and Flush Status Register, FFSR* on page 9-585
0x0304	FFCR	RW	0x00000000	32	*Formatter and Flush Control Register, FFCR* on page 9-586
0x0308	PSCR	RW	0x0000000a	32	*Periodic Synchronization Counter Register, PSCR* on page 9-589
0x0EE0	ITEVTINTR	WO	0x00000000	32	*Integration Test Event and Interrupt Control Register, ITEVTINTR* on page 9-590
0x0EE8	ITTRFLIN	RO	0x00000000	32	*Integration Test Trigger In and Flush In register, ITTRFLIN* on page 9-591
0x0EEC	ITATBDATA0	RO	0x00000000	32	*Integration Test ATB Data 0 Register, ITATBDATA0* on page 9-592
0x0EF0	ITATBCTR2	WO	0x00000000	32	*Integration Test ATB Control 2 Register, ITATBCTR2* on page 9-594
0x0EF4	ITATBCTR1	RO	0x00000000	32	*Integration Test ATB Control 1 Register, ITATBCTR1* on page 9-595
0x0EF8	ITATBCTR0	RO	0x00000000	32	*Integration Test ATB Control 0 Register, ITATBCTR0* on page 9-596
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-597
0x0FA0	CLAIMSET	RW	0x0000000f	32	*Claim Tag Set Register, CLAIMSET* on page 9-598
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-599
0x0FB8	AUTHSTATUS	RO	0x00000000	32	*Authentication Status Register, AUTHSTATUS* on page 9-600
0x0FC8	DEVID	RO	0x04010-c0	32	*Device Configuration Register, DEVID* on page 9-602

Table 9-425 css600_tmc_ets - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FC4	DEVID1	RO	0x00000001	32	*Device Configuration Register, DEVID1* on page 9-603
0x0FCC	DEVTYPE	RO	0x00000021	32	*Device Type Identifier Register, DEVTYPE* on page 9-604
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-605
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-606
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-607
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-608
0x0FE0	PIDR0	RO	0x000000e8	32	*Peripheral Identification Register 0, PIDR0* on page 9-609
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-610
0x0FE8	PIDR2	RO	0x00000001	32	*Peripheral Identification Register 2, PIDR2* on page 9-611
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-612
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-613
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-614
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-615
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-616

9.15.2 Register descriptions

This section describes the `css600_tmc_ets` registers.

9.15.1 Register summary on page 9-576 provides cross references to individual registers.

RAM Size register, RSZ

Defines the size of trace memory in units of 32-bit words.

The RSZ register characteristics are:

Attributes

Offset 0x0004
Type Read-write
Reset 0x0000000-
Width 32

The following figure shows the bit assignments.

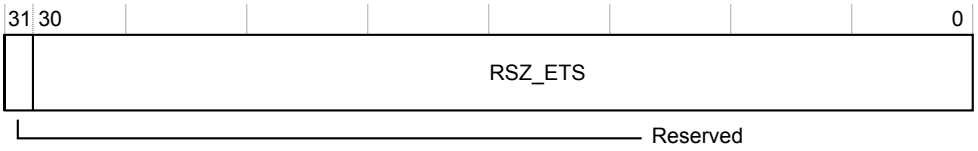

Figure 9-412 RSZ register bit assignments

The following table shows the bit assignments.

Table 9-426 RSZ register bit assignments

Bits	Reset value	Name	Function
[30:0]	IMPLEMENTATION_DEFINED	RSZ_ETS	RAM Size. Indicates the size of trace memory in 32-bit words. Trace memory size is fixed as one AXI Stream data word.

Status register, STS

Indicates the status of the Trace Memory Controller. After a reset, software must ignore all the fields of this register except STS.TMCReady. The other fields have meaning only when the TMC has left the Disabled state. Writes to all RO fields of this register are ignored.

The STS register characteristics are:

Attributes

Offset	`0x000c`
Type	Read-write
Reset	`0x000000--`
Width	32

The following figure shows the bit assignments.

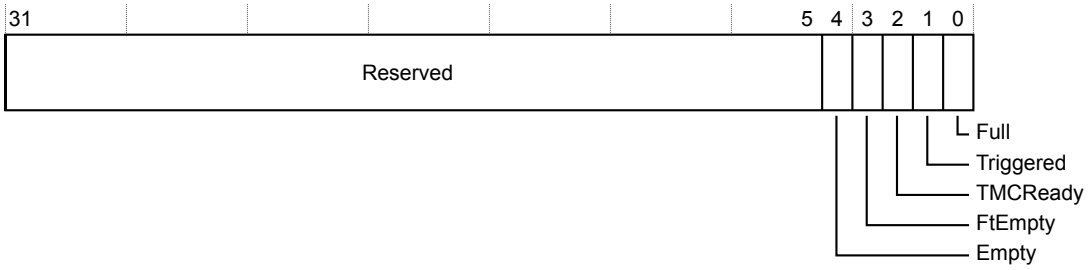

Figure 9-413 STS register bit assignments

The following table shows the bit assignments.

Table 9-427 STS register bit assignments

Bits	Reset value	Name	Function
[4]	UNKNOWN	Empty	Trace buffer empty. If set, this bit indicates that the trace memory does not contain any valid trace data. However, this does not mean that the pipeline stages within the TMC are empty. To determine whether the internal pipeline stages are empty, the software must read the STS.TMCReady bit.
[3]	UNKNOWN	FtEmpty	Formatter pipeline empty. This bit is deprecated and is present in this register to support backwards compatibility with earlier versions of the ETB. It is set when trace capture has stopped, and all internal pipelines and buffers have drained. Unlike STS.TMCReady, it is not affected by buffer drains and AXI accesses. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.
[2]	`0b1`	TMCReady	TMC ready. This flag is set when trace capture has stopped and all internal pipelines and buffers have drained, the TMC is not draining as a result of FFCR.DrainBuffer bit being set (ETF only), and the AXI interface is not busy and the response for final AXI write has been received (ETR only) are all true. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state.
[1]	UNKNOWN	Triggered	TMC triggered. This bit is set when trace capture is in progress and the TMC has detected a trigger event. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. A trigger event is when the TMC has written a set number of data words, as programmed in the TRG register, into the trace memory after a rising edge of **trigin** input, or a trigger packet (**atid_s** = `0x7D`) is received in the input trace.
[0]	UNKNOWN	Full	Trace memory full. This bit helps in determining the amount of valid data present in the trace memory. Writes to this bit are allowed in Disabled state. However, it is not affected by the reprogramming of pointer registers in this state.

RAM Read Data register, RRD

Reading this register allows data to be read from the trace memory. When the memory width is greater than the APB width (32-bit), multiple reads to the same location must be performed to get the complete memory dataword. The RRP register is incremented only after the last of the required number of read accesses to fetch one memory dataword completes.

The RRD register characteristics are:

Attributes

Offset `0x0010`
Type Read-only
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

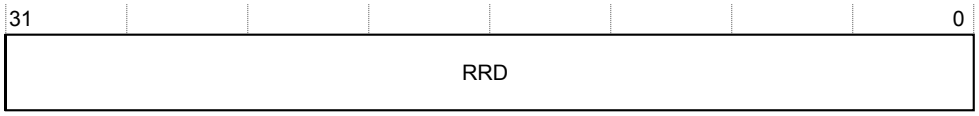

Figure 9-414 RRD register bit assignments

The following table shows the bit assignments.

Table 9-428 RRD register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	RRD	Returns the data read from trace memory.

Trigger Counter register, TRG

In Circular Buffer mode, the Trigger Counter register specifies the number of 32-bit words to capture in the trace memory, after detection of either a rising edge on the **trigin** input or a trigger packet in the incoming trace stream, that is, where **atid_s** = `0x7D`. The value programmed must be aligned to the frame length of 128 bits.

The TRG register characteristics are:

Attributes

Offset `0x001c`
Type Read-write
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

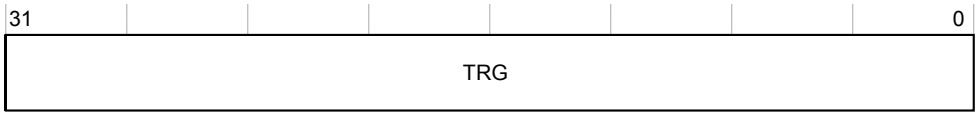

Figure 9-415 TRG register bit assignments

The following table shows the bit assignments.

Table 9-429 TRG register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	TRG	Trigger count. This count represents the number of 32-bit words of trace that are captured between a trigger packet and a trigger event. On writes, the value written must be aligned to the frame length of 128 bits.

Control Register, CTL

This register controls trace stream capture. Setting the CTL.TraceCaptEn bit to 1 enables the TMC to capture the trace data. When trace capture is enabled, the formatter behavior is controlled by the FFCR register.

The CTL register characteristics are:

Attributes

Offset `0x0020`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

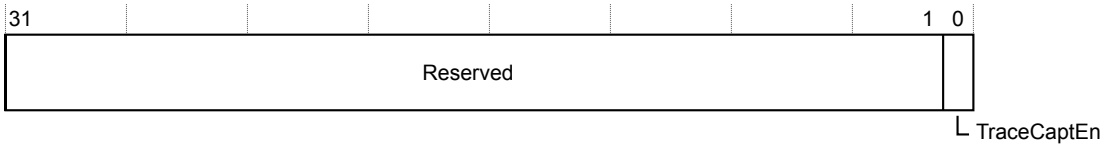

Figure 9-416 CTL register bit assignments

The following table shows the bit assignments.

Table 9-430 CTL register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	TraceCaptEn	Trace capture enable. **0** Disable trace capture. **1** Enable trace capture.

RAM Write Data register, RWD

When in Disabled state, a write to this register stores data at the location pointed to by the RWP/RWPHI registers. When not in Disabled state, writes are ignored. When the memory data width, as indicated by the DEVID.MEM_WIDTH register field, is greater than 32 bits, multiple writes to this register must be performed together to write a full memory width of data. For example, if the memory width is 128 bits, then writes to this register must be performed 4 at a time. When a full memory width of data has been written to this register, the data is written to the trace memory and the RWP is incremented to the next memory word.

The RWD register characteristics are:

Attributes

Offset `0x0024`
Type Write-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

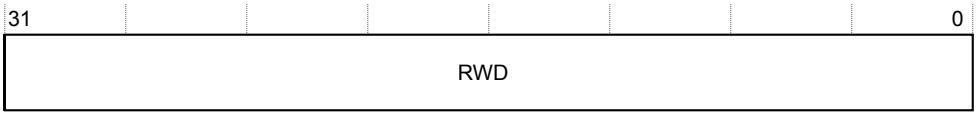

Figure 9-417 RWD register bit assignments

The following table shows the bit assignments.

Table 9-431 RWD register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	RWD	Data written to this register is placed in the trace memory.

Mode register, MODE

This register controls the TMC operating mode. The operating mode can only be changed when the TMC is in Disabled state, that is, when CTL.TraceCaptEn = 0. Attempting to write to this register in any other state results in UNPREDICTABLE behavior. The operating mode is ignored when in Disabled state.

The MODE register characteristics are:

Attributes

Offset `0x0028`
Type Read-write
Reset `0x0000000-`
Width 32

The following figure shows the bit assignments.

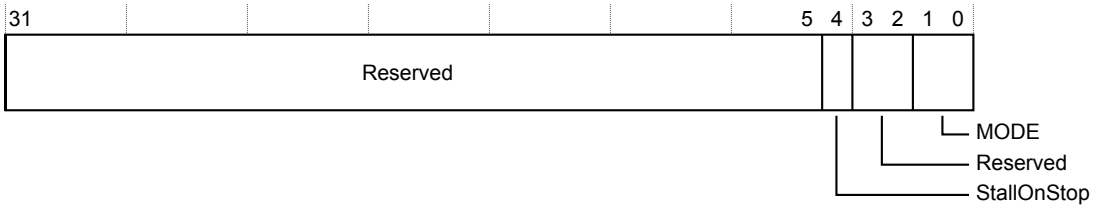

Figure 9-418 MODE register bit assignments

The following table shows the bit assignments.

Table 9-432 MODE register bit assignments

Bits	Reset value	Name	Function
[4]	`0b0`	StallOnStop	Stall On Stop. If this bit is set and the formatter stops as a result of a stop event, the output **atready_s** is de-asserted to stall the ATB interface and avoid loss of trace. If this bit is clear and the formatter stops as a result of a stop event, signal **atready_s** remains asserted but the TMC discards further incoming trace.
[1:0]	UNKNOWN	MODE	Fixed to 0 since TMC always operates in Circular Buffer mode in this configuration.

Formatter and Flush Status Register, FFSR

This register indicates the status of the Formatter, and the status of Flush request.

The FFSR register characteristics are:

Attributes

Offset 0x0300
Type Read-only
Reset 0x0000000-
Width 32

The following figure shows the bit assignments.

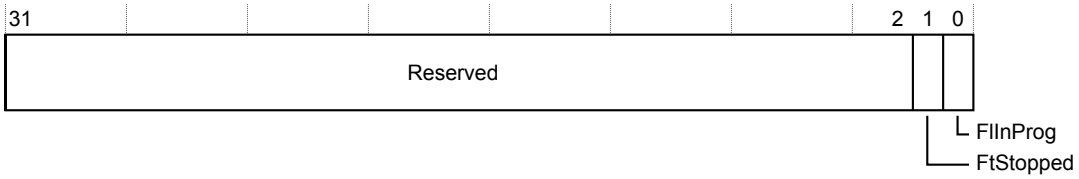

Figure 9-419 FFSR register bit assignments

The following table shows the bit assignments.

Table 9-433 FFSR register bit assignments

Bits	Reset value	Name	Function
[1]	UNKNOWN	FtStopped	Formatter Stopped. This bit behaves the same way as STS.FtEmpty. It is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. The FFCR.FtStopped bit is deprecated and is present in this register to support backwards-compatibility with earlier versions of the ETB. **0** Trace capture has not yet completed. **1** Trace capture has completed and all captured trace data has been written to the trace memory.
[0]	UNKNOWN	FlInProg	Flush In Progress. This bit indicates whether the TMC is currently processing a flush request. In the ETB and ETR configurations, the flush initiation is controlled by the flush control bits in the FFCR register. In the ETF configuration, the flush request can also be from the ATB Master port. This bit is cleared to 0 when leaving the Disabled state and retains its value when entering the Disabled state. When in Disabled state, this bit is not updated. **0** No flush activity in progress. **1** Flush in progress on the ATB slave interface or the TMC internal pipeline.

Formatter and Flush Control Register, FFCR

This register allows user control of the stop, trigger, and flush events.

The FFCR register characteristics are:

Attributes

Offset `0x0304`

Type Read-write

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

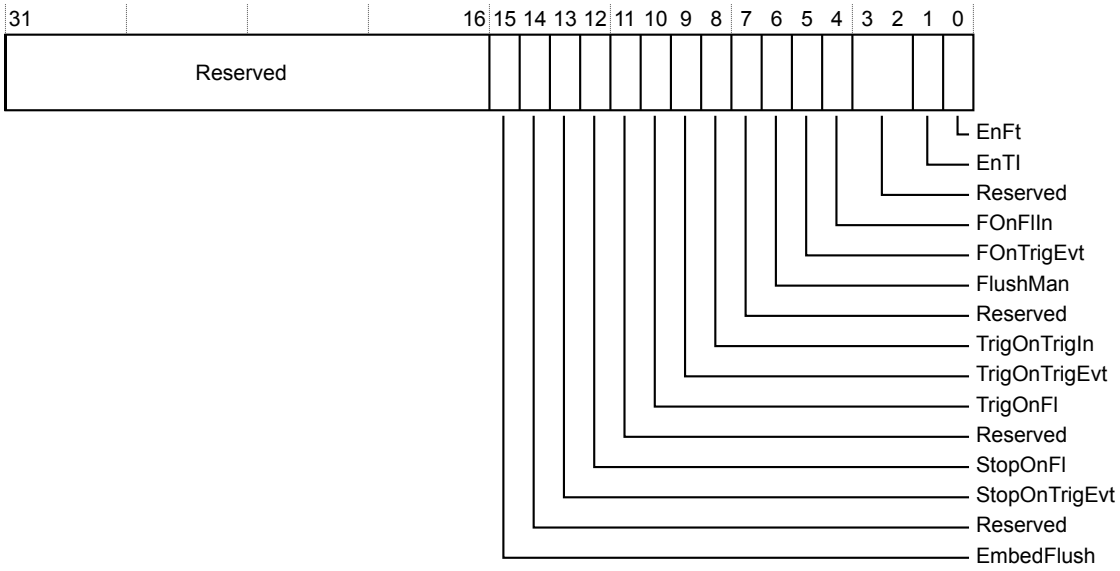

Figure 9-420 FFCR register bit assignments

The following table shows the bit assignments.

Table 9-434 FFCR register bit assignments

Bits	Reset value	Name	Function
[15]	`0b0`	EmbedFlush	Embed Flush ID (flush completion packet). Enables insertion of Flush ID `0x7B` with a single byte of data payload = `0x00` in the output trace, immediately after the last flush data byte, when a flush completes on the ATB slave interface. This bit is effective only in Normal formatting modes. In Bypass mode, the Flush ID insertion remains disabled and this bit is ignored. **0** Disable Flush ID insertion. **1** Enable Flush ID insertion.
[13]	`0b0`	StopOnTrigEvt	Stop On Trigger Event. If this bit is set, the formatter is stopped when a Trigger Event has been observed. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.

Table 9-434 FFCR register bit assignments (continued)

Bits	Reset value	Name	Function
[12]	0b0	StopOnFl	Stop On Flush. If this bit is set, the formatter is stopped on completion of a flush operation. The initiation of a flush operation is controlled by programming the register bits FFCR.FlushMan, FFCR.FOnTrigEvt, and FFCR.FOnFlIn. When a flush-initiation condition occurs, **afvalid_s** is asserted, and when the flush completion is received, that is, **afready_s**=1, trace capture is stopped. Any remaining data in the formatter is appended with a post-amble and written to trace memory. The flush operation is then complete. When the TMC is configured as an ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a formatter stop regardless of the value that is programmed in this bit.
[10]	0b0	TrigOnFl	Indicate on trace stream the completion of flush. If this bit is set, a trigger is indicated on the trace stream when **afready_s** is received for a flush in progress. If this bit is clear, no triggers are embedded in the trace stream on flush completion. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. When the TMC is configured as ETF, if a flush is initiated by the ATB Master interface, its completion does not lead to a trigger indication on the trace stream regardless of the value that is programmed in this bit.
[9]	0b0	TrigOnTrigEvt	Indicate on trace stream the occurrence of a Trigger Event. If this bit is set, a trigger is indicated on the output trace stream when a Trigger Event occurs. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[8]	0b0	TrigOnTrigIn	Indicate on trace stream the occurrence of a rising edge on **trigin**. If this bit is set, a trigger is indicated on the trace stream when a rising edge is detected on the **trigin** input. If Trigger Insertion is disabled, that is, FFCR.EnTI=0, then trigger indication on the trace stream is blocked regardless of the value that is programmed in this bit.
[6]	0b0	FlushMan	Manually generate a flush of the system. Writing 1 to this bit causes a flush to be generated. This bit is cleared automatically when, in formatter bypass mode, **afready_s** was sampled high, or, in normal formatting mode, **afready_s** was sampled high and all flush data was output to the trace memory. If CTL.TraceCaptEn=0, writes to this bit are ignored.
[5]	0b0	FOnTrigEvt	Flush on Trigger Event. Setting this bit generates a flush when a Trigger Event occurs. If FFCR.StopOnTrigEvt is set, this bit is ignored. This bit must be used only in CB mode because in FIFO modes, the TMC is a trace link rather than a trace sink and trigger events are related to trace sink functionality. If trace capture is enabled in SWF1, SWF2, or HWF mode with this bit set, it results in UNPREDICTABLE behavior.
[4]	0b0	FOnFlIn	Setting this bit enables the detection of transitions on the **flushin** input by the TMC. If this bit is set and the formatter has not already stopped, a rising edge on **flushin** initiates a flush request.

Table 9-434 FFCR register bit assignments (continued)

Bits	Reset value	Name	Function
[1]	`0b0`	EnTI	Enable Trigger Insertion. Setting this bit enables the insertion of triggers in the formatted trace stream. A trigger is indicated by inserting one byte of data `0x00` with **atid_s**=`0x7D` in the trace stream. Trigger indication on the trace stream is also controlled by the register bits FFCR.TrigOnFl, FFCR.TrigOnTrigEvt, and FFCR.TrigOnTrigIn. This bit can only be changed when the TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled.
[0]	`0b0`	EnFt	Enable Formatter. If this bit is set, formatting is enabled. This bit takes effect when not in Disabled state, and is ignored in Disabled state. If this bit is clear, formatting is disabled. In this case, the incoming trace data is assumed to be from a single trace source. If multiple trace IDs are received by the TMC when trace capture is enabled and the formatter is disabled, it results in interleaving of trace data. Disabling of formatting is deprecated, and is supported in the TMC for backwards-compatibility with earlier versions of the ETB. Therefore, disabling of formatting is supported only in CB mode. Features in the TMC such as the FIFO modes and the FFCR.DrainBuffer bit that are not part of the earlier versions of the ETB do not support disabling of formatting. This bit can only be changed when TMC is in Disabled state. If FFCR.EnTI bit is set when FFCR.EnFt is 0, it results in formatting being enabled. If the TMC is enabled in a mode other than Circular Buffer mode with formatting disabled, it results in formatting being enabled.

Periodic Synchronization Counter Register, PSCR

This register determines the reload value of the Periodic Synchronization Counter. This counter enables the frequency of sync packets to be optimized to the trace capture buffer size. The default behavior of the counter is to generate periodic synchronization requests, **syncreq_s**, on the ATB slave interface.

The PSCR register characteristics are:

Attributes

Offset	`0x0308`
Type	Read-write
Reset	`0x0000000a`
Width	32

The following figure shows the bit assignments.

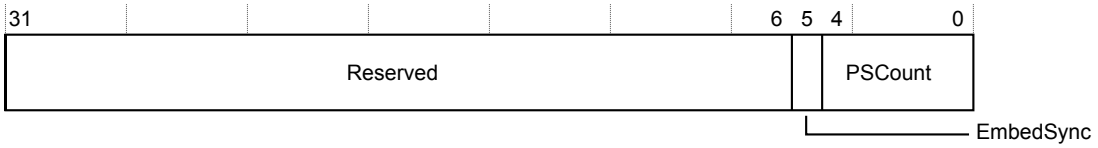

Figure 9-421 PSCR register bit assignments

The following table shows the bit assignments.

Table 9-435 PSCR register bit assignments

Bits	Reset value	Name	Function
[5]	`0b0`	EmbedSync	Embed Frame Sync Packet in the trace stream. Setting this bit to 1 enables the formatter to insert frame sync packets in the trace stream at periodic intervals. If this bit is set and the Synchronization Counter is enabled, the formatter inserts a 32-bit frame sync packet in the trace stream when the counter reaches 0. This bit is effective only when formatting is enabled, that is when FFCR.EnTI=1 or FFCR.EnFt=1, and it is ignored when the formatter is in bypass mode.
[4:0]	`0b01010`	PSCount	Periodic Synchronization Count. Determines the reload value of the Synchronization Counter. The reload value takes effect the next time the counter reaches zero. When trace capture is enabled, the Synchronization Counter counts the number of bytes of trace data that is stored into the trace memory, regardless of whether the trace data has been formatted by the TMC or not, since the occurrence of the last sync request on the ATB slave interface. When the counter reaches 0, a sync request is sent on the ATB slave interface. Reads from this register return the reload value that is programmed in this register. This field resets to `0x0A`, that is, the default sync period is 2^10 bytes. If a reserved value is programmed in this register field, the value `0x1B` is used instead, and subsequent reads from this register also return `0x1B`. The following constraints apply to the values written to the PSCount field: `0x0` - synchronization is disabled, `0x1-0x6` - reserved, `0x7-0x1B` - synchronization period is 2^PSCount bytes. The smallest value `0x7` gives a sync period of 128 bytes. The maximum allowed value `0x1B` gives a sync period of 2^27 bytes, `0x1C-0x1F` - reserved.

Integration Test Event and Interrupt Control Register, ITEVTINTR

This register controls the values of event and interrupt outputs in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITEVTINTR register characteristics are:

Attributes

Offset	`0x0ee0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

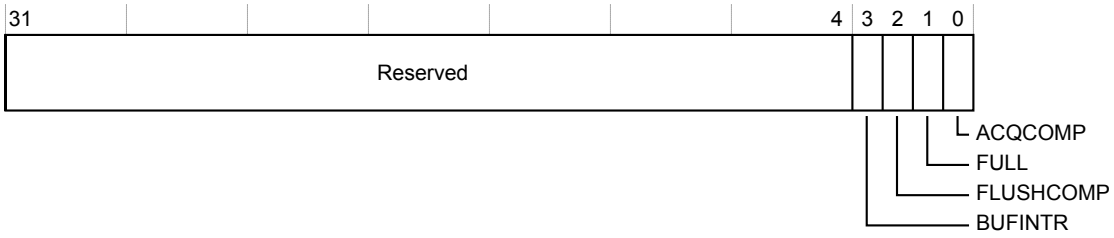

Figure 9-422 ITEVTINTR register bit assignments

The following table shows the bit assignments.

Table 9-436 ITEVTINTR register bit assignments

Bits	Reset value	Name	Function
[3]	`0b0`	BUFINTR	Controls the value of **bufintr** output in integration mode.
[2]	`0b0`	FLUSHCOMP	Controls the value of **flushcomp** output in integration mode.
[1]	`0b0`	FULL	Controls the value of **full** output in integration mode.
[0]	`0b0`	ACQCOMP	Controls the value of **acqcomp** output in integration mode.

Integration Test Trigger In and Flush In register, ITTRFLIN

This register captures the values of the **flushin** and **trigin** inputs in integration mode. In functional mode, this register behaves as RAZ/WI.

The ITTRFLIN register characteristics are:

Attributes

Offset	`0x0ee8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

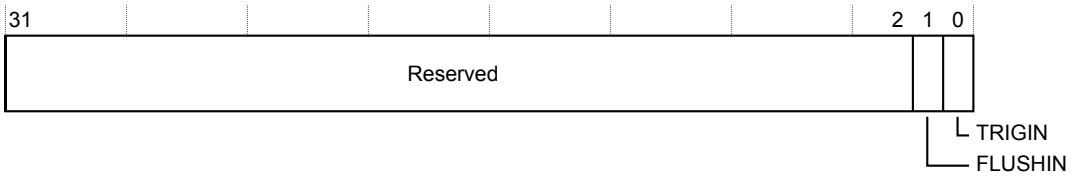

Figure 9-423 ITTRFLIN register bit assignments

The following table shows the bit assignments.

Table 9-437 ITTRFLIN register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	FLUSHIN	Integration status of **flushin** input. In integration mode, this bit latches to 1 on a rising edge of the **flushin** input. It is cleared when the register is read or when integration mode is disabled.
[0]	`0b0`	TRIGIN	Integration status of **trigin** input. In integration mode, this bit latches to 1 on a rising edge of the **trigin** input. It is cleared when the register is read or when integration mode is disabled.

Integration Test ATB Data 0 Register, ITATBDATA0

This register captures the value of **atdata_s** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding **atdata_s** bits. The width of this register is given by: 1+(ATB DATA WIDTH)/8.

The ITATBDATA0 register characteristics are:

Attributes

Offset `0x0eec`
Type Read-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

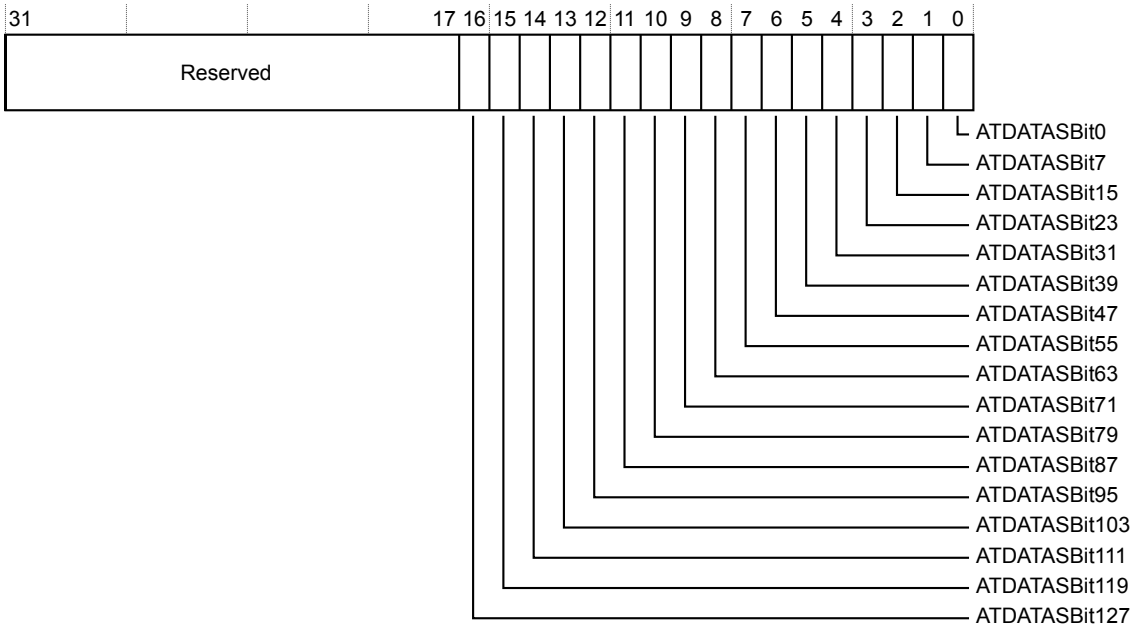

Figure 9-424 ITATBDATA0 register bit assignments

The following table shows the bit assignments.

Table 9-438 ITATBDATA0 register bit assignments

Bits	Reset value	Name	Function
[16]	`0b0`	ATDATASBit127	Returns the value of **atdata_s[127]** input in integration mode.
[15]	`0b0`	ATDATASBit119	Returns the value of **atdata_s[119]** input in integration mode.
[14]	`0b0`	ATDATASBit111	Returns the value of **atdata_s[111]** input in integration mode.
[13]	`0b0`	ATDATASBit103	Returns the value of **atdata_s[103]** input in integration mode.
[12]	`0b0`	ATDATASBit95	Returns the value of **atdata_s[95]** input in integration mode.
[11]	`0b0`	ATDATASBit87	Returns the value of **atdata_s[87]** input in integration mode.
[10]	`0b0`	ATDATASBit79	Returns the value of **atdata_s[79]** input in integration mode.
[9]	`0b0`	ATDATASBit71	Returns the value of **atdata_s[71]** input in integration mode.
[8]	`0b0`	ATDATASBit63	Returns the value of **atdata_s[63]** input in integration mode.

Table 9-438 ITATBDATA0 register bit assignments (continued)

Bits	Reset value	Name	Function
[7]	0b0	ATDATASBit55	Returns the value of **atdata_s[55]** input in integration mode.
[6]	0b0	ATDATASBit47	Returns the value of **atdata_s[47]** input in integration mode.
[5]	0b0	ATDATASBit39	Returns the value of **atdata_s[39]** input in integration mode.
[4]	0b0	ATDATASBit31	Returns the value of **atdata_s[31]** input in integration mode.
[3]	0b0	ATDATASBit23	Returns the value of **atdata_s[23]** input in integration mode.
[2]	0b0	ATDATASBit15	Returns the value of **atdata_s[15]** input in integration mode.
[1]	0b0	ATDATASBit7	Returns the value of **atdata_s[7]** input in integration mode.
[0]	0b0	ATDATASBit0	Returns the value of **atdata_s[0]** input in integration mode.

Integration Test ATB Control 2 Register, ITATBCTR2

This register enables control of ATB slave outputs **atready_s**, **afvalid_s**, and **syncreq_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, the value that is written to any bit of this register is driven on the output pin that is controlled by that bit and the reads return `0x0`.

The ITATBCTR2 register characteristics are:

Attributes

Offset	`0x0ef0`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

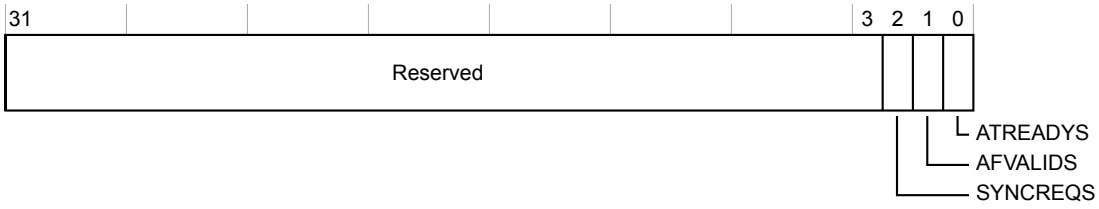

Figure 9-425 ITATBCTR2 register bit assignments

The following table shows the bit assignments.

Table 9-439 ITATBCTR2 register bit assignments

Bits	Reset value	Name	Function
[2]	`0b0`	SYNCREQS	Controls the value of **syncreq_s** output in integration mode.
[1]	`0b0`	AFVALIDS	Controls the value of **afvalid_s** output in integration mode.
[0]	`0b0`	ATREADYS	Controls the value of **atready_s** output in integration mode.

Integration Test ATB Control 1 Register, ITATBCTR1

This register captures the value of the **atid_s[6:0]** input in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of **atid_s** input.

The ITATBCTR1 register characteristics are:

Attributes

Offset `0x0ef4`

Type Read-only

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

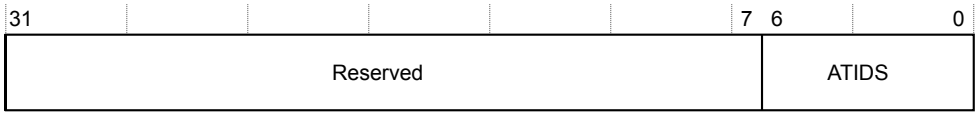

Figure 9-426 ITATBCTR1 register bit assignments

The following table shows the bit assignments.

Table 9-440 ITATBCTR1 register bit assignments

Bits	Reset value	Name	Function
[6:0]	`0b0000000`	ATIDS	Returns the value of **atid_s[6:0]** input in integration mode.

Integration Test ATB Control 0 Register, ITATBCTR0

This register captures the values of ATB slave inputs **atvalid_s**, **afready_s**, **atwakeup_s**, and **atbytes_s** in integration mode. In functional mode, this register behaves as RAZ/WI. In integration mode, writes to this register are ignored and the reads return the value of corresponding input pins. The width of this register is given by: $8+\log_2$(ATB DATA WIDTH/8).

The ITATBCTR0 register characteristics are:

Attributes

Offset	`0x0ef8`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

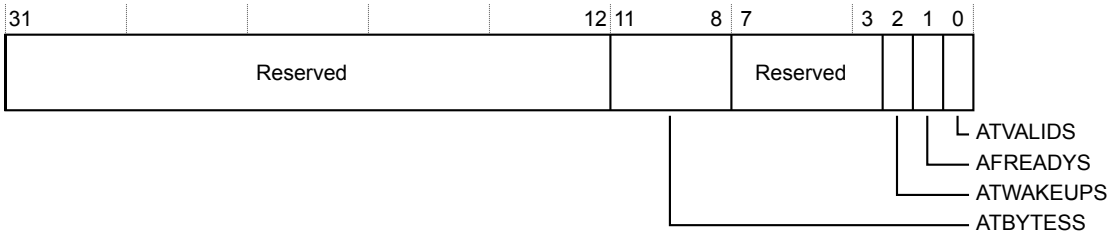

Figure 9-427 ITATBCTR0 register bit assignments

The following table shows the bit assignments.

Table 9-441 ITATBCTR0 register bit assignments

Bits	Reset value	Name	Function
[11:8]	`0b0000`	ATBYTESS	Returns the value of **atbytes_s** input in integration mode. N=8+$\log_2$(ATB DATA WIDTH/8).
[2]	`0b0`	ATWAKEUPS	Returns the value of **atwakeup_s** input in integration mode.
[1]	`0b0`	AFREADYS	Returns the value of **afready_s** input in integration mode.
[0]	`0b0`	ATVALIDS	Returns the value of **atvalid_s** input in integration mode.

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.

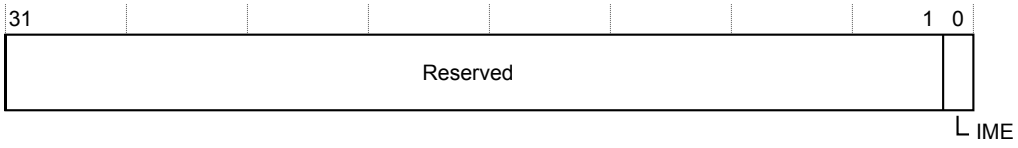

Figure 9-428 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-442 ITCTRL register bit assignments

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

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 `0x0000000f`

Width 32

The following figure shows the bit assignments.

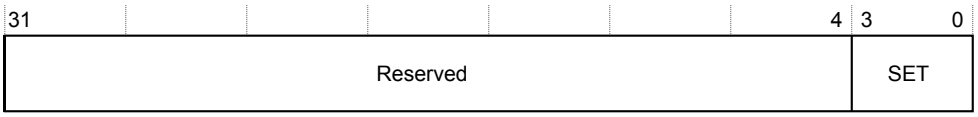

Figure 9-429 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-443 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	SET	A bit programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

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.

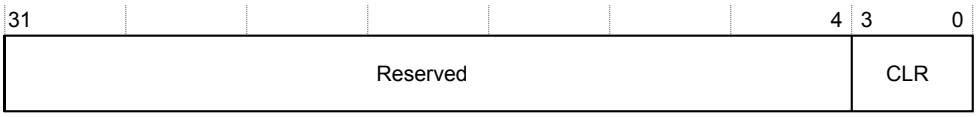

Figure 9-430 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-444 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	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.

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	`0x00000000`
Width	32

The following figure shows the bit assignments.

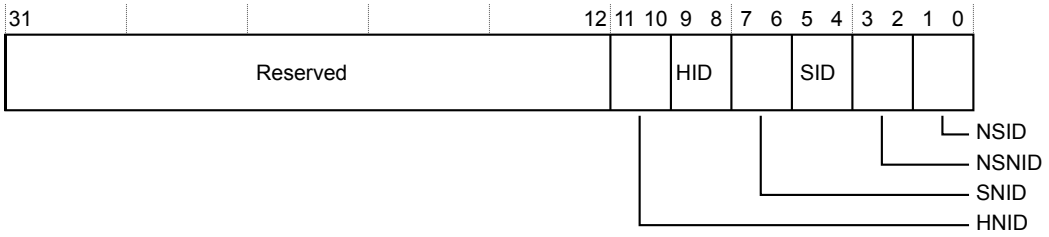

Figure 9-431 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-445 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	`0b00`	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-445 AUTHSTATUS register bit assignments (continued)

Bits	Reset value	Name	Function
[3:2]	0b00	NSNID	Non-secure non-invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.
[1:0]	0b00	NSID	Non-secure invasive debug. 0x0 Functionality not implemented or controlled elsewhere. 0x1 Reserved. 0x2 Functionality disabled. 0x3 Functionality enabled.

Device Configuration Register, DEVID

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

The DEVID register characteristics are:

Attributes

Offset `0x0fc8`
Type Read-only
Reset `0x04010-c0`
Width 32

The following figure shows the bit assignments.

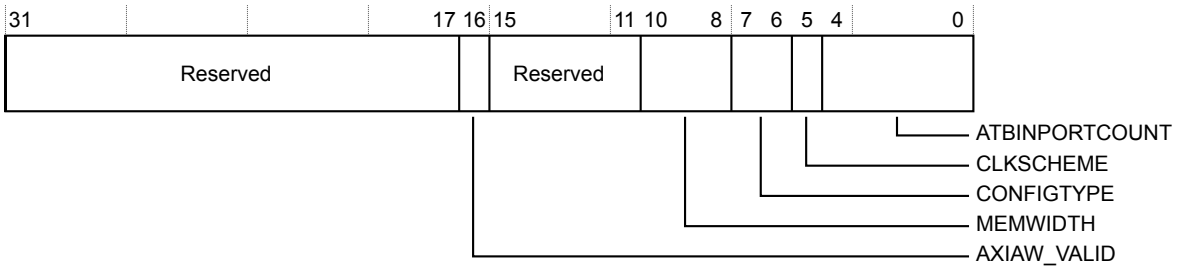

Figure 9-432 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-446 DEVID register bit assignments

Bits	Reset value	Name	Function
[16]	`0b1`	AXIAW_VALID	Indicates whether field DEVID.AW is valid.
[10:8]	IMPLEMENTATION_DEFINED	MEMWIDTH	This value is equal to `ATB_DATA_WIDTH`. `0x2` Memory interface databus is 32-bits wide. `0x3` Memory interface databus is 64-bits wide. `0x4` Memory interface databus is 128-bits wide. `0x5` Memory interface databus is 256-bits wide.
[7:6]	`0b11`	CONFIGTYPE	Returns `0x3`, indicating ETS configuration.
[5]	`0b0`	CLKSCHEME	RAM Clocking Scheme. This value indicates the TMC RAM clocking scheme used, that is, whether the TMC RAM operates synchronously or asynchronously to the TMC clock. Fixed to 0 indicating that TMC RAM clock is synchronous to the **clk** input.
[4:0]	`0b00000`	ATBINPORTCOUNT	Hidden Level of ATB input multiplexing. This value indicates the type/number of ATB multiplexing present on the input ATB. Fixed to `0x00` indicating that no multiplexing is present.

Device Configuration Register, DEVID1

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

The DEVID1 register characteristics are:

Attributes

Offset `0x0fc4`

Type Read-only

Reset `0x00000001`

Width 32

The following figure shows the bit assignments.

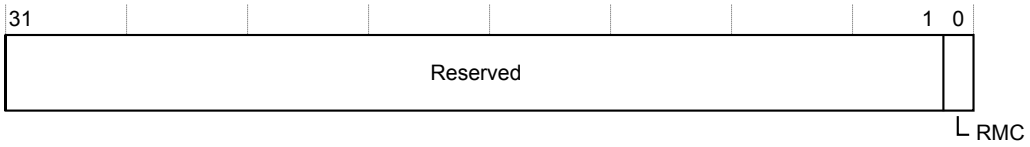

Figure 9-433 DEVID1 register bit assignments

The following table shows the bit assignments.

Table 9-447 DEVID1 register bit assignments

Bits	Reset value	Name	Function
[31:1]	`0x00000000`	-	Reserved.
[0]	1	RMC	Indicates register management mode 1 is implemented.

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 `0x00000021`
Width 32

The following figure shows the bit assignments.

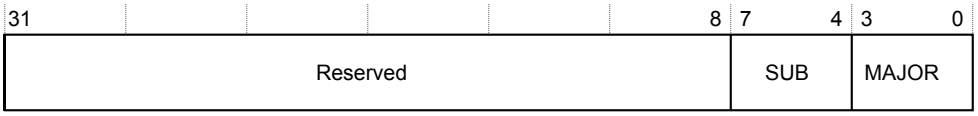

Figure 9-434 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-448 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0010`	SUB	Minor classification. Returns `0x2`, indicating this component is a Buffer.
[3:0]	`0b0001`	MAJOR	Major classification. Returns `0x1`, indicating this component is a Trace Sink.

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.

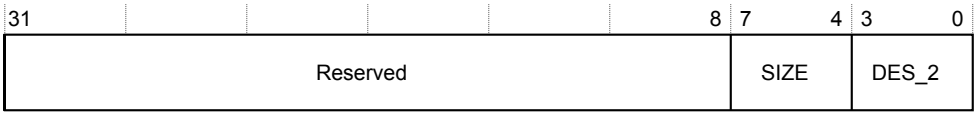

Figure 9-435 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-449 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

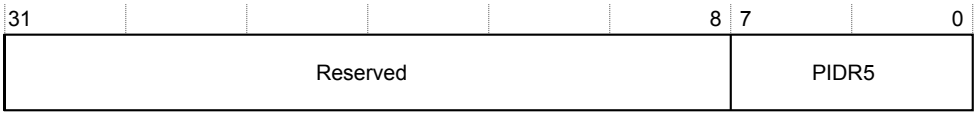

Figure 9-436 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-450 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

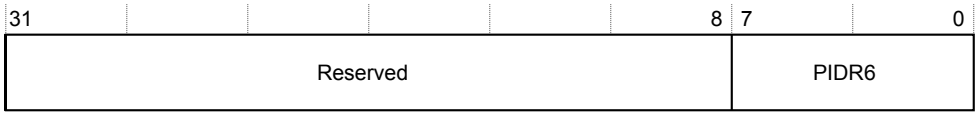

Figure 9-437 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-451 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

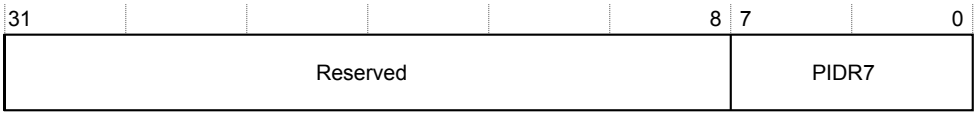

Figure 9-438 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-452 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000e8`

Width 32

The following figure shows the bit assignments.

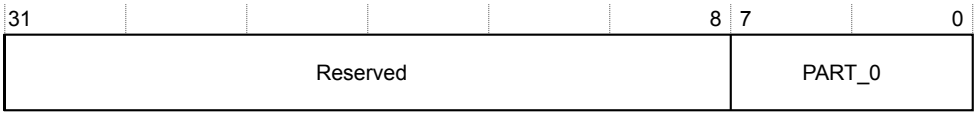

Figure 9-439 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-453 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11101000`	PART_0	Part number (lower 8 bits). Returns `0xe8`, indicating TMC ETR/ETS.

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.

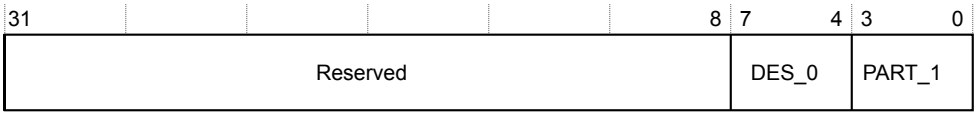

Figure 9-440 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-454 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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 `0x00000001`
Width 32

The following figure shows the bit assignments.

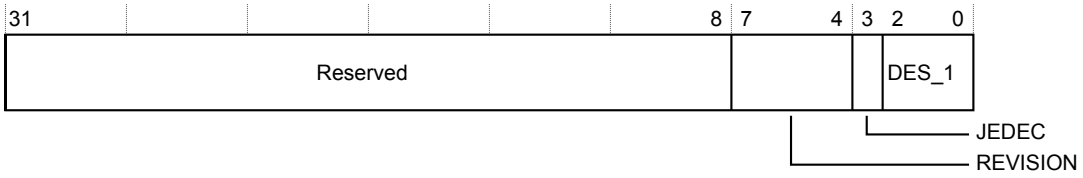

Figure 9-441 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-455 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0x1`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

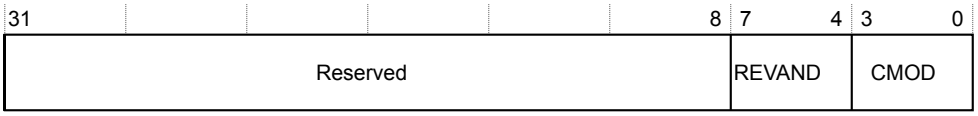

Figure 9-442 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-456 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

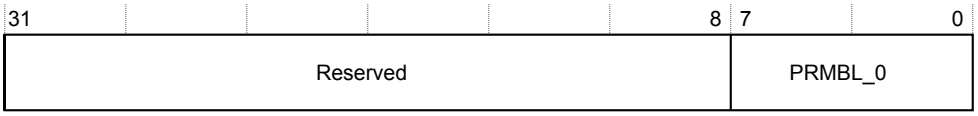

Figure 9-443 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-457 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

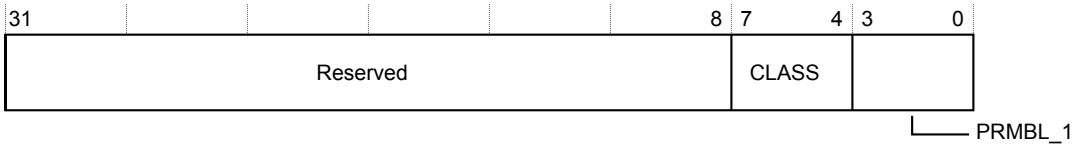

Figure 9-444 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-458 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

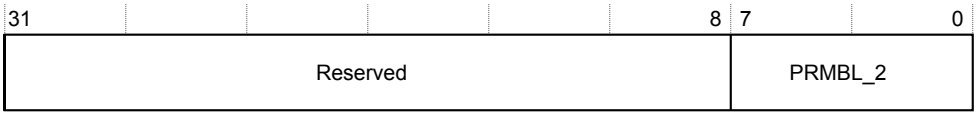

Figure 9-445 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-459 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

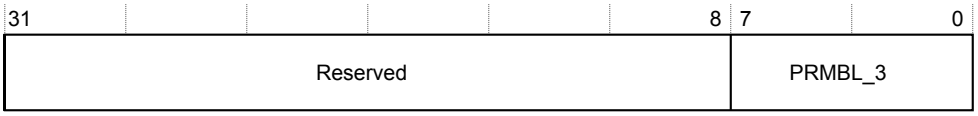

Figure 9-446 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-460 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.16 css600_tsgen introduction

This section describes the functions and programmers model of the `css600_tsgen`.

Note

The `css600_tsgen` has two interfaces - a control interface, refered to in the following sections as `APB4_Slave_0`, and a read-only interface, refered to as `APB4_Slave_1`.

This section contains the following subsections:

- *9.16.1 Register summary* on page 9-617.
- *9.16.2 Control interface register descriptions* on page 9-618.
- *9.16.3 Register summary* on page 9-637.
- *9.16.4 Read-only interface register descriptions* on page 9-638.

9.16.1 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.

Table 9-461 css600_tsgen - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	CNTCR	RW	0x00000000	32	*Counter Control Register, CNTCR* on page 9-619
0x0004	CNTSR	RO	0x00000000	32	*Counter Status Register, CNTSR* on page 9-620
0x0008	CNTCVL	RW	0x00000000	32	*Current value of Counter[31:0], CNTCVL* on page 9-621
0x000C	CNTCVU	RW	0x00000000	32	*Current value of Counter[63:32], CNTCVU* on page 9-622
0x0020	CNTFID0	RW	0x00000000	32	*Base Frequency ID register, CNTFID0* on page 9-623
0x0EF8	ITSTAT	RO	0x0000000-	32	*Integration Test Status Register, ITSTAT* on page 9-624
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-625
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-626
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-627
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-628
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-629
0x0FE0	PIDR0	RO	0x00000093	32	*Peripheral Identification Register 0, PIDR0* on page 9-630
0x0FE4	PIDR1	RO	0x000000b1	32	*Peripheral Identification Register 1, PIDR1* on page 9-631
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-632
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-633
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-634
0x0FF4	CIDR1	RO	0x000000f0	32	*Component Identification Register 1, CIDR1* on page 9-635

Table 9-461 css600_tsgen - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-636
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-637

9.16.2 Control interface register descriptions

This section describes the `css600_tsgen` control interface registers.

9.16.1 Register summary on page 9-617 provides cross references to individual registers.

9.16.3 Register summary on page 9-637 provides cross references to individual registers.

Counter Control Register, CNTCR

The counter control register controls the counter increments.

The CNTCR register characteristics are:

Attributes

Offset	0x0000
Type	Read-write
Reset	0x00000000
Width	32

The following figure shows the bit assignments.

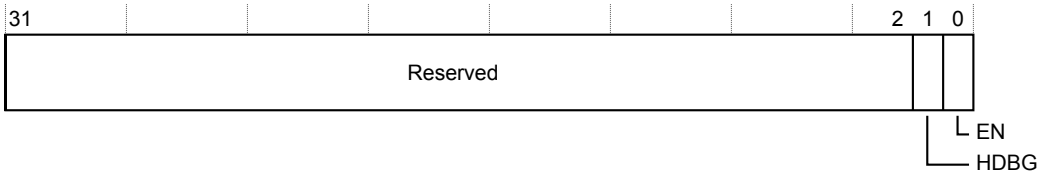

Figure 9-447 CNTCR register bit assignments

The following table shows the bit assignments.

Table 9-462 CNTCR register bit assignments

Bits	Reset value	Name	Function
[1]	0b0	HDBG	Halt On Debug. **0** Do not halt on debug. The **halt_req** signal into the counter has no effect. **1** Halt on debug. When the **halt_req** pulse is received, the count value is held static.
[0]	0b0	EN	Enable Bit. **0** The counter is disabled. Count is not incrementing. **1** The counter is enabled. Count is incrementing.

Counter Status Register, CNTSR

Identifies the status of the counter.

The CNTSR register characteristics are:

Attributes

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

The following figure shows the bit assignments.

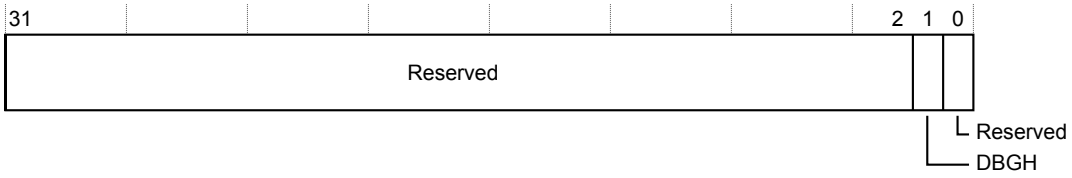

Figure 9-448 CNTSR register bit assignments

The following table shows the bit assignments.

Table 9-463 CNTSR register bit assignments

Bits	Reset value	Name	Function
[1]	`0b0`	DBGH	Debug status. **0** Debug is halted **1** Debug is not halted.

Current value of Counter[31:0], CNTCVL

Reads or writes the lower 32 bits of the current counter value.

The CNTCVL register characteristics are:

Attributes

Offset `0x0008`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

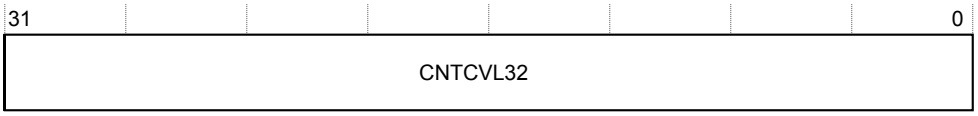

Figure 9-449 CNTCVL register bit assignments

The following table shows the bit assignments.

Table 9-464 CNTCVL register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	CNTCVL32	Reads to this register return the lower 32 bits of the current timestamp counter value. To change the current timestamp value, write the lower 32 bits of the new value to this register before writing the upper 32 bits to CNTCVU. The timestamp value is not changed until the CNTCVU register is written to.

Current value of Counter[63:32], CNTCVU

Reads or writes the upper 32 bits of the current counter value. The control interface must clear the CNTCR.EN bit or set CNTCR.HDBG and hlt_dbg asserted on the input to stop the counter, before writing to this register.

The CNTCVU register characteristics are:

Attributes

Offset `0x000c`

Type Read-write

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

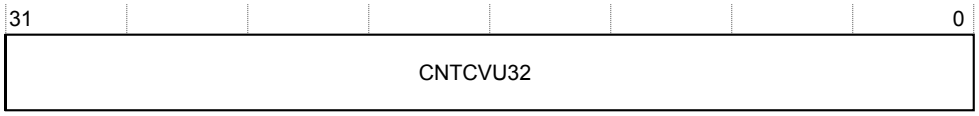

Figure 9-450 CNTCVU register bit assignments

The following table shows the bit assignments.

Table 9-465 CNTCVU register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	CNTCVU32	Reads to this register return the upper 32 bits of the current timestamp counter value. To change the current timestamp value, write the lower 32 bits of the new value to CNTCVL before writing the upper 32 bits to this register. The 64-bit timestamp value is updated with the value from both writes when this register is written to.

Base Frequency ID register, CNTFID0

You must program this register to match the clock frequency of the timestamp generator, in ticks per second. For example, for a 50 MHz clock, program `0x02FAF080`. The real-time speed of the counter does not depend on the value of this register. This register reports, to the reader, the speed of the counter as programmed by the system firmware.

The CNTFID0 register characteristics are:

Attributes

Offset `0x0020`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

Figure 9-451 CNTFID0 register bit assignments

The following table shows the bit assignments.

Table 9-466 CNTFID0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	Freq	Frequency in number of ticks per second. Up to 4GHz can be specified.

Integration Test Status Register, ITSTAT

Integration test register to view **halt_req** and **restart_req** values.

The ITSTAT register characteristics are:

Attributes

Offset	`0x0ef8`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

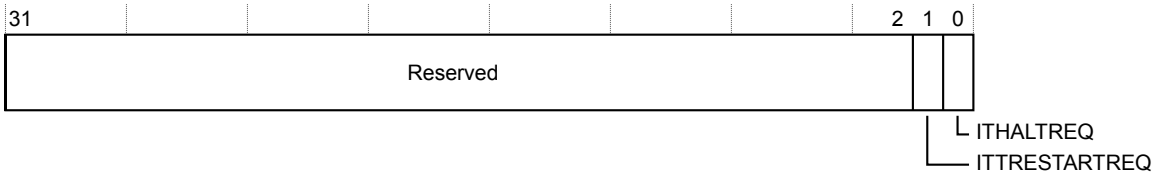

Figure 9-452 ITSTAT register bit assignments

The following table shows the bit assignments.

Table 9-467 ITSTAT register bit assignments

Bits	Reset value	Name	Function
[1]	UNKNOWN	ITTRESTARTREQ	Integration Test Restart Request status of the **restart_req** input. Integration testing mode: Behaves as a sticky bit and latches to 1 when tsgen receives restart request. Cleared on reading this register. If **restart_req** is asserted in the same cycle as an APB read of this register, the read takes priority and the register is cleared as a result. Always returns 0 in normal functional mode.
[0]	UNKNOWN	ITHALTREQ	Integration Test Halt Request status of the **halt_req** input. Integration testing mode: Behaves as a sticky bit and latches to 1 when tsgen receives halt request. Cleared on reading this register. If **halt_req** is asserted in the same cycle as an APB read of this register, the read takes priority and the register is cleared as a result. Always returns 0 in normal functional mode.

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.

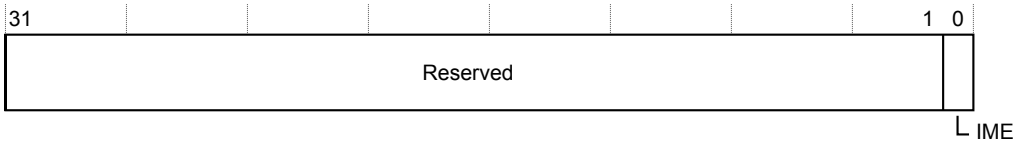

Figure 9-453 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-468 ITCTRL register bit assignments

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

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.

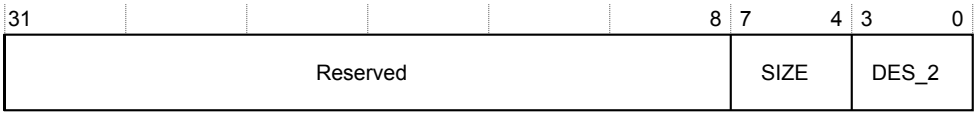

Figure 9-454 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-469 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

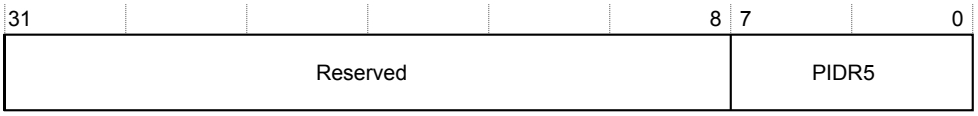

Figure 9-455 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-470 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

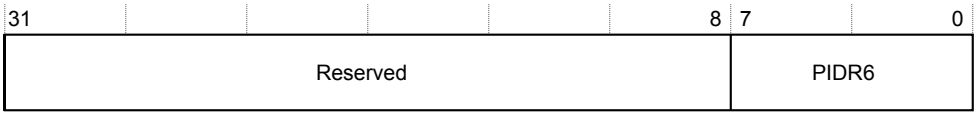

Figure 9-456 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-471 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

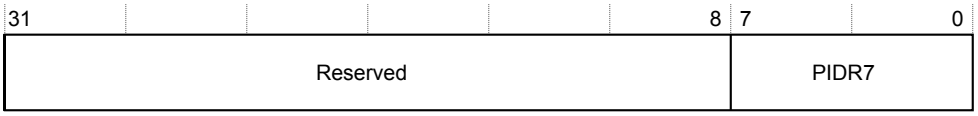

Figure 9-457 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-472 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x00000093`

Width 32

The following figure shows the bit assignments.

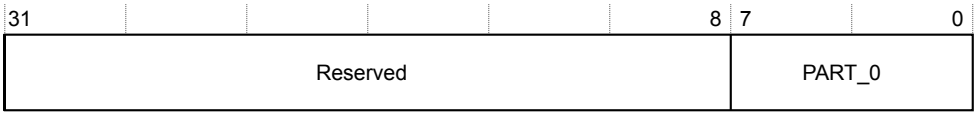

Figure 9-458 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-473 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10010011`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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 `0x000000b1`

Width 32

The following figure shows the bit assignments.

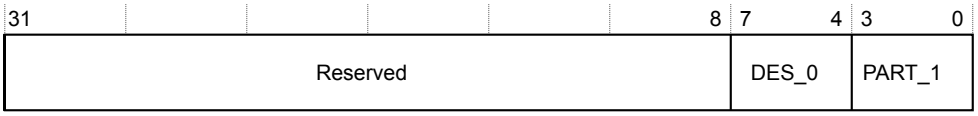

Figure 9-459 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-474 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]	`0b0001`	PART_1	Part number, bits[11:8]. This is selected by the designer of the component.

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 `0x0000000b`
Width 32

The following figure shows the bit assignments.

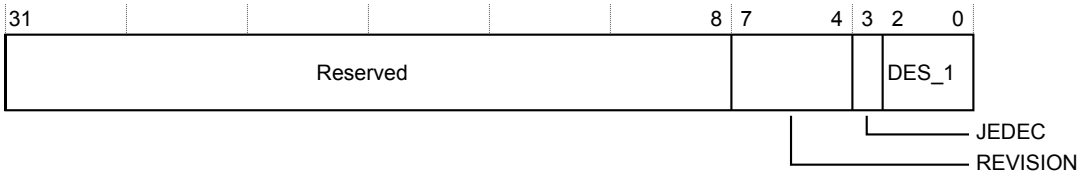

Figure 9-460 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-475 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

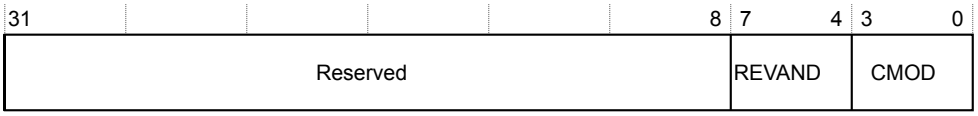

Figure 9-461 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-476 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

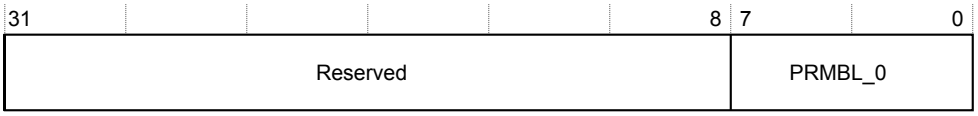

Figure 9-462 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-477 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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 `0x000000f0`

Width 32

The following figure shows the bit assignments.

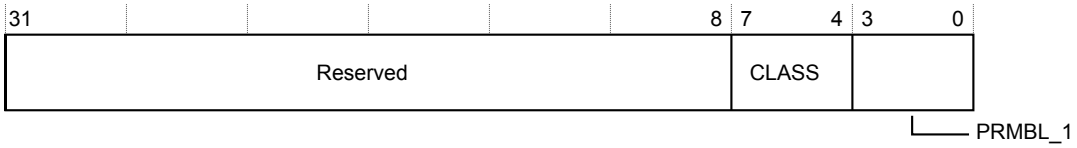

Figure 9-463 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-478 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1111`	CLASS	Component class. Returns `0xf`, indicating CoreLink, PrimeCell, or system component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

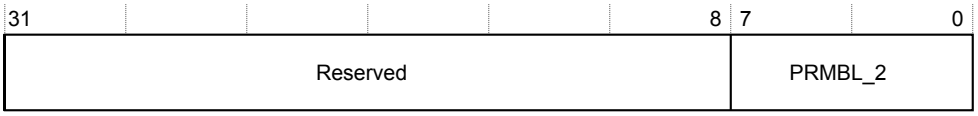

Figure 9-464 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-479 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

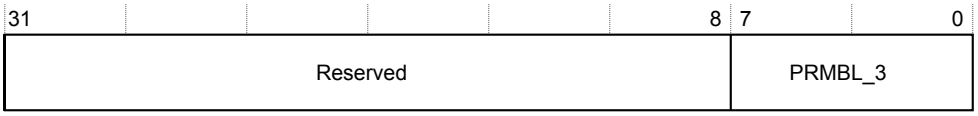

Figure 9-465 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-480 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.16.3 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 9-481 css600_tsgen - APB4_Slave_1 register summary

Offset	Name	Type	Reset	Width	Description
`0x0000`	CNTCVLREAD	RO	`0x00000000`	32	*Current value of Counter[31:0], CNTCVLREAD* on page 9-639
`0x0004`	CNTCVUREAD	RO	`0x00000000`	32	*Current value of Counter[63:32], CNTCVUREAD* on page 9-640
`0x0FD0`	PIDR4	RO	`0x00000004`	32	*Peripheral Identification Register 4, PIDR4* on page 9-641
`0x0FD4`	PIDR5	RO	`0x00000000`	32	*Peripheral Identification Register 5, PIDR5* on page 9-642
`0x0FD8`	PIDR6	RO	`0x00000000`	32	*Peripheral Identification Register 6, PIDR6* on page 9-643
`0x0FDC`	PIDR7	RO	`0x00000000`	32	*Peripheral Identification Register 7, PIDR7* on page 9-644
`0x0FE0`	PIDR0	RO	`0x00000093`	32	*Peripheral Identification Register 0, PIDR0* on page 9-645
`0x0FE4`	PIDR1	RO	`0x000000b1`	32	*Peripheral Identification Register 1, PIDR1* on page 9-646
`0x0FE8`	PIDR2	RO	`0x0000000b`	32	*Peripheral Identification Register 2, PIDR2* on page 9-647
`0x0FEC`	PIDR3	RO	`0x00000000`	32	*Peripheral Identification Register 3, PIDR3* on page 9-648
`0x0FF0`	CIDR0	RO	`0x0000000d`	32	*Component Identification Register 0, CIDR0* on page 9-649

Table 9-481 css600_tsgen - APB4_Slave_1 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0FF4	CIDR1	RO	0x000000f0	32	*Component Identification Register 1, CIDR1* on page 9-650
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-651
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-652

9.16.4 Read-only interface register descriptions

This section describes the `css600_tsgen` read-only interface registers.

9.16.1 Register summary on page 9-617 provides cross references to individual registers.

9.16.3 Register summary on page 9-637 provides cross references to individual registers.

Current value of Counter[31:0], CNTCVLREAD

Reads the lower 32 bits of the current counter value.

The CNTCVLREAD register characteristics are:

Attributes

Offset 0x0000

Type Read-only

Reset 0x00000000

Width 32

The following figure shows the bit assignments.

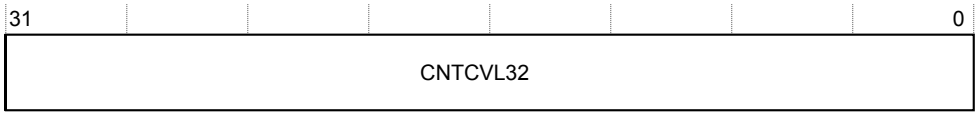

Figure 9-466 CNTCVLREAD register bit assignments

The following table shows the bit assignments.

Table 9-482 CNTCVLREAD register bit assignments

Bits	Reset value	Name	Function
[31:0]	0x0	CNTCVL32	The lower 32 bits of the current timestamp counter value.

Current value of Counter[63:32], CNTCVUREAD

Reads the upper 32 bits of the current counter value.

The CNTCVUREAD register characteristics are:

Attributes

Offset 0x0004
Type Read-only
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

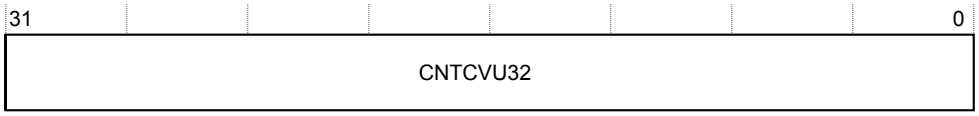

Figure 9-467 CNTCVUREAD register bit assignments

The following table shows the bit assignments.

Table 9-483 CNTCVUREAD register bit assignments

Bits	Reset value	Name	Function
[31:0]	0x0	CNTCVU32	The upper 32 bits of the current timestamp counter value.

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.

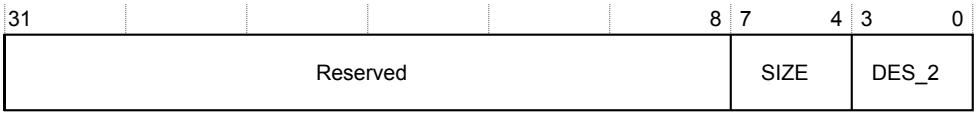

Figure 9-468 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-484 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

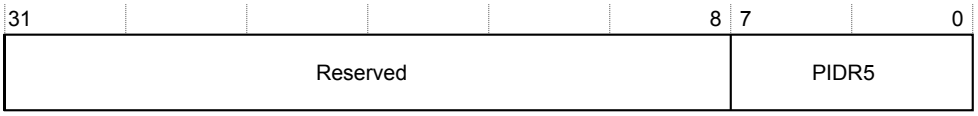

Figure 9-469 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-485 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

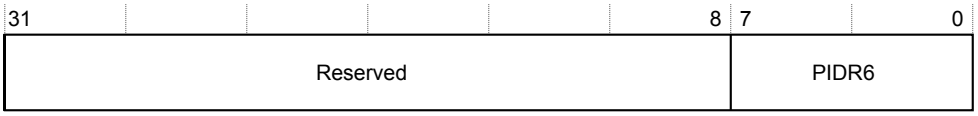

Figure 9-470 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-486 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

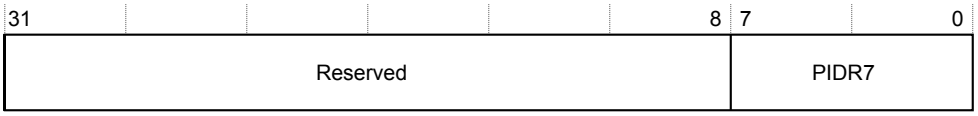

Figure 9-471 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-487 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x00000093`

Width 32

The following figure shows the bit assignments.

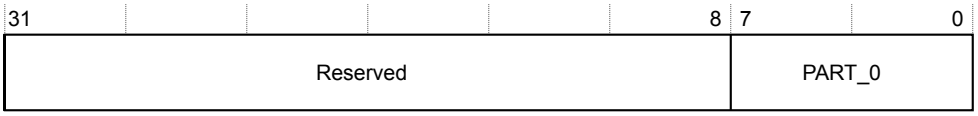

Figure 9-472 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-488 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10010011`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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 `0x000000b1`

Width 32

The following figure shows the bit assignments.

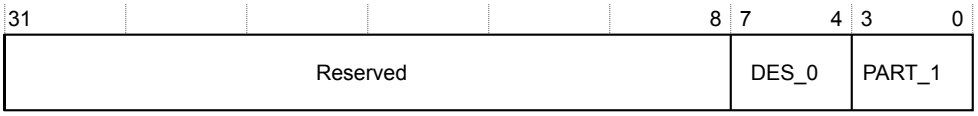

Figure 9-473 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-489 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]	`0b0001`	PART_1	Part number, bits[11:8]. This is selected by the designer of the component.

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	`0x0000000b`
Width	32

The following figure shows the bit assignments.

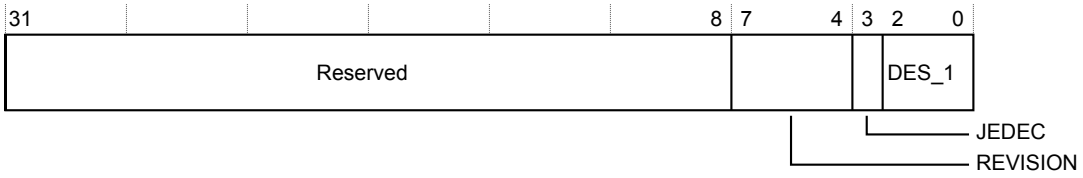

Figure 9-474 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-490 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

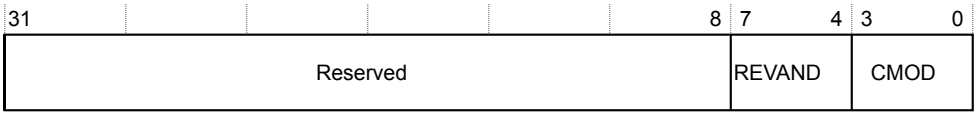

Figure 9-475 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-491 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

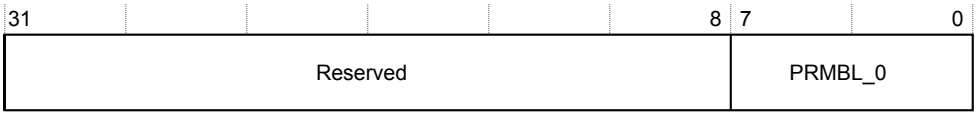

Figure 9-476 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-492 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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 `0x000000f0`

Width 32

The following figure shows the bit assignments.

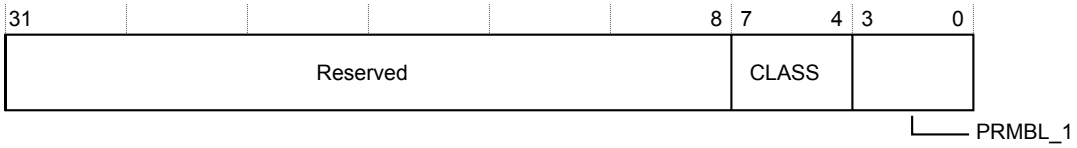

Figure 9-477 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-493 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1111`	CLASS	Component class. Returns `0xf`, indicating CoreLink, PrimeCell, or system component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

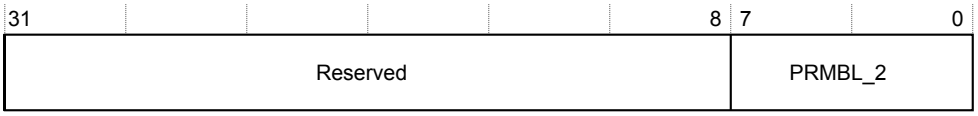

Figure 9-478 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-494 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

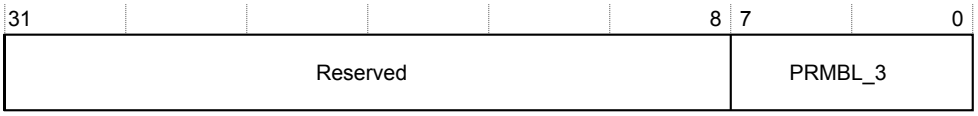

Figure 9-479 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-495 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

9.17 css600_cti introduction

This section describes the functions and programmers model of the `css600_cti`.

This section contains the following subsections:

9.17.1 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.

Table 9-496 css600_cti - APB4_Slave_0 register summary

Offset	Name	Type	Reset	Width	Description
0x0000	CTICONTROL	RW	0x00000000	32	*CTI Control register, CTICONTROL* on page 9-656
0x0010	CTIINTACK	WO	0x00000000	32	*CTI Interrupt Acknowledge register, CTIINTACK* on page 9-657
0x0014	CTIAPPSET	RW	0x00000000	32	*CTI Application Channel Set register, CTIAPPSET* on page 9-658
0x0018	CTIAPPCLEAR	WO	0x00000000	32	*CTI Application Channel Clear register, CTIAPPCLEAR* on page 9-659
0x001C	CTIAPPPULSE	WO	0x00000000	32	*CTI Application Channel Pulse register, CTIAPPPULSE* on page 9-660
0x0020	CTIINEN0	RW	0x00000000	32	*CTI Trigger 0 to Channel Enable register, CTIINEN0* on page 9-661
0x0024	CTIINEN1	RW	0x00000000	32	*CTI Trigger 1 to Channel Enable register, CTIINEN1* on page 9-662
0x0028	CTIINEN2	RW	0x00000000	32	*CTI Trigger 2 to Channel Enable register, CTIINEN2* on page 9-663
…	…	…	…	…	…
0x009C	CTIINEN31	RW	0x00000000	32	*CTI Trigger 31 to Channel Enable register, CTIINEN31* on page 9-664
0x00A0	CTIOUTEN0	RW	0x00000000	32	*CTI Channel to Trigger 0 Enable register, CTIOUTEN0* on page 9-665
0x00A4	CTIOUTEN1	RW	0x00000000	32	*CTI Channel to Trigger 1 Enable register, CTIOUTEN1* on page 9-666
0x00A8	CTIOUTEN2	RW	0x00000000	32	*CTI Channel to Trigger 2 Enable register, CTIOUTEN2* on page 9-667
…	…	…	…	…	…
0x011C	CTIOUTEN31	RW	0x00000000	32	*CTI Channel to Trigger 31 Enable register, CTIOUTEN31* on page 9-668

Table 9-496 css600_cti - APB4_Slave_0 register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x0130	CTITRIGINSTATUS	RO	0x--------	32	*CTI Trigger Input Status register, CTITRIGINSTATUS* on page 9-669
0x0134	CTITRIGOUTSTATUS	RO	0x--------	32	*CTI Trigger Output Status register, CTITRIGOUTSTATUS* on page 9-670
0x0138	CTICHINSTATUS	RO	0x0000000-	32	*CTI Channel Input Status register, CTICHINSTATUS* on page 9-671
0x013C	CTICHOUTSTATUS	RO	0x0000000-	32	*CTI Channel Output Status register, CTICHOUTSTATUS* on page 9-672
0x0140	CTIGATE	RW	0x0000000f	32	*Enable CTI Channel Gate register, CTIGATE* on page 9-673
0x0144	ASICCTRL	RW	0x00000000	32	*External Multiplexer Control register, ASICCTRL* on page 9-674
0x0EE4	ITCHOUT	RW	0x00000000	32	*Integration Test Channel Output register, ITCHOUT* on page 9-675
0x0EE8	ITTRIGOUT	RW	0x00000000	32	*Integration Test Trigger Output register, ITTRIGOUT* on page 9-676
0x0EF4	ITCHIN	RO	0x00000000	32	*Integration Test Channel Input register, ITCHIN* on page 9-677
0x0EF8	ITTRIGIN	RO	0x00000000	32	*Integration Test Trigger Input register, ITTRIGIN* on page 9-678
0x0F00	ITCTRL	RW	0x00000000	32	*Integration Mode Control Register, ITCTRL* on page 9-679
0x0FA0	CLAIMSET	RW	0x0000000f	32	*Claim Tag Set Register, CLAIMSET* on page 9-680
0x0FA4	CLAIMCLR	RW	0x00000000	32	*Claim Tag Clear Register, CLAIMCLR* on page 9-681
0x0FA8	DEVAFF0	RO	0x--------	32	*Device Affinity register 0, DEVAFF0* on page 9-682
0x0FAC	DEVAFF1	RO	0x--------	32	*Device Affinity register 1, DEVAFF1* on page 9-683
0x0FB8	AUTHSTATUS	RO	0x0000000-	32	*Authentication Status Register, AUTHSTATUS* on page 9-684
0x0FBC	DEVARCH	RO	0x47701a14	32	*Device Architecture Register, DEVARCH* on page 9-686
0x0FC8	DEVID	RO	0x01040100	32	*Device Configuration Register, DEVID* on page 9-687
0x0FCC	DEVTYPE	RO	0x00000014	32	*Device Type Identifier Register, DEVTYPE* on page 9-688
0x0FD0	PIDR4	RO	0x00000004	32	*Peripheral Identification Register 4, PIDR4* on page 9-689
0x0FD4	PIDR5	RO	0x00000000	32	*Peripheral Identification Register 5, PIDR5* on page 9-690
0x0FD8	PIDR6	RO	0x00000000	32	*Peripheral Identification Register 6, PIDR6* on page 9-691
0x0FDC	PIDR7	RO	0x00000000	32	*Peripheral Identification Register 7, PIDR7* on page 9-692
0x0FE0	PIDR0	RO	0x000000ed	32	*Peripheral Identification Register 0, PIDR0* on page 9-693
0x0FE4	PIDR1	RO	0x000000b9	32	*Peripheral Identification Register 1, PIDR1* on page 9-694
0x0FE8	PIDR2	RO	0x0000000b	32	*Peripheral Identification Register 2, PIDR2* on page 9-695
0x0FEC	PIDR3	RO	0x00000000	32	*Peripheral Identification Register 3, PIDR3* on page 9-696
0x0FF0	CIDR0	RO	0x0000000d	32	*Component Identification Register 0, CIDR0* on page 9-697
0x0FF4	CIDR1	RO	0x00000090	32	*Component Identification Register 1, CIDR1* on page 9-698
0x0FF8	CIDR2	RO	0x00000005	32	*Component Identification Register 2, CIDR2* on page 9-699
0x0FFC	CIDR3	RO	0x000000b1	32	*Component Identification Register 3, CIDR3* on page 9-700

9.17.2 Register descriptions

This section describes the `css600_cti` registers.

9.17.1 Register summary on page 9-653 provides cross references to individual registers.

Note

The number of bits implemented in the CTIINTACK, CTITRIGINSTATUS, CTITRIGOUTSTATUS, ITTRIGOUT, and ITTRIGIN registers depend on the number of triggers <n> implemented. There is 1 bit in each register for each implemented trigger, bits[<n>-1:0], with bits[31:<n>] Reserved.

Registers CTIINEN0..CTIINEN31 and CTIOUTEN0..CTIOUTEN31 are all implemented, regardless of the number of triggers implemented. Reads from registers that are associated with unimplemented triggers return UNDEFINED values, and writes have no effect.

CTI Control register, CTICONTROL

The CTI control register enables and disables the CTI.

The CTICONTROL register characteristics are:

Attributes

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

The following figure shows the bit assignments.

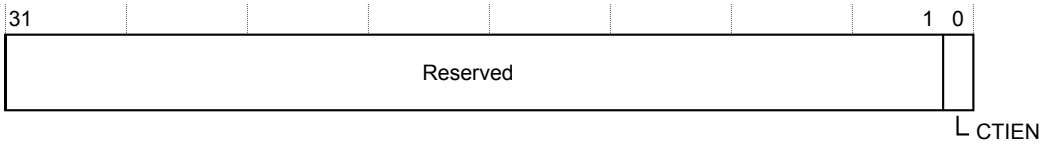

Figure 9-480 CTICONTROL register bit assignments

The following table shows the bit assignments.

Table 9-497 CTICONTROL register bit assignments

Bits	Reset value	Name	Function
[0]	`0b0`	CTIEN	Enable control. **0** CTI disabled. **1** CTI enabled.

CTI Interrupt Acknowledge register, CTIINTACK

Software acknowledge for trigger outputs. The CTIINTACK register is a bit map that allows selective clearing of trigger output events. If the SW_HANDSHAKE parameter for a trigger output is set, indicating that the output latches HIGH when an event is sent to that output, then the output remains HIGH until the corresponding INTACK bit is written to a 1. A write of a bit to 0 has no effect. This allows different software threads to be responsible for clearing different trigger outputs without needing to perform a read-modify-write operation to find which bits are set.

The CTIINTACK register characteristics are:

Attributes

Offset 0x0010
Type Write-only
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

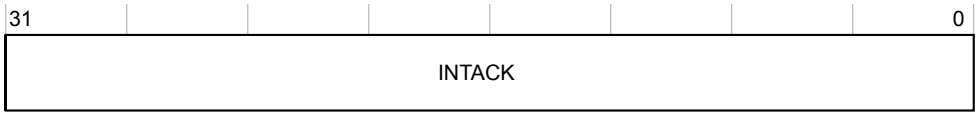

Figure 9-481 CTIINTACK register bit assignments

The following table shows the bit assignments.

Table 9-498 CTIINTACK register bit assignments

Bits	Reset value	Name	Function
[31:0]	0x0	INTACK	Acknowledges the corresponding ctitrigout output.

CTI Application Channel Set register, CTIAPPSET

The application channel set register allows software to set any channel output. This register can be used by software to generate a channel event in place of a hardware source on a trigger input. In a system where all events are sent as single cycle pulses, this register must not be used. It is only retained for compatibility with older systems and software. The register is implemented before the CTIGATE register so, for the channel event to propagate outside the CTI, it is necessary for the corresponding CTIGATE bit to be 1.

The CTIAPPSET register characteristics are:

Attributes

Offset `0x0014`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

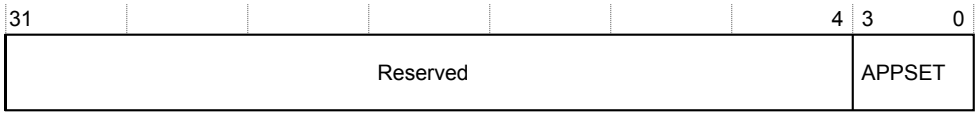

Figure 9-482 CTIAPPSET register bit assignments

The following table shows the bit assignments.

Table 9-499 CTIAPPSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	APPSET	Sets the corresponding internal channel flag. **0** Read: application channel is inactive. Write: has no effect. **1** Read: application channel is active. Write: sets the channel output.

CTI Application Channel Clear register, CTIAPPCLEAR

The application channel clear register allows software to clear any channel output. This register can be used by software to clear a channel event in place of a hardware source on a trigger input. In a system where all events are sent as single cycle pulses, this register must not be used. It is only retained for compatibility with older systems and software. The register is implemented before the CTIGATE register so, for the channel event to propagate outside the CTI, it is necessary for the corresponding CTIGATE bit to be 1.

The CTIAPPCLEAR register characteristics are:

Attributes

Offset `0x0018`
Type Write-only
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

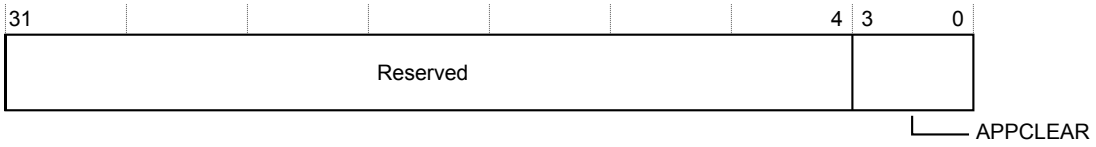

Figure 9-483 CTIAPPCLEAR register bit assignments

The following table shows the bit assignments.

Table 9-500 CTIAPPCLEAR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	APPCLEAR	Clears the corresponding internal channel flag. **0** No effect. **1** Clears the channel output.

CTI Application Channel Pulse register, CTIAPPPULSE

The application channel pulse register allows software to pulse any channel output. This register can be used by software to pulse a channel event in place of a hardware source on a trigger input. The register is implemented before the CTIGATE register so, for the channel event to propagate outside the CTI, it is necessary for the corresponding CTIGATE bit to be 1.

The CTIAPPPULSE register characteristics are:

Attributes

Offset	`0x001c`
Type	Write-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

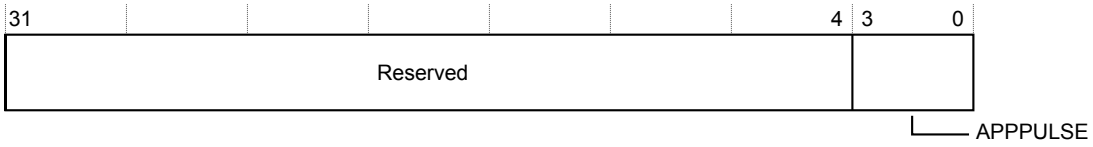

Figure 9-484 CTIAPPPULSE register bit assignments

The following table shows the bit assignments.

Table 9-501 CTIAPPPULSE register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	APPPULSE	Pulses the channel outputs. **0** No effect. **1** Pulse channel event for one **clk** cycle.

CTI Trigger 0 to Channel Enable register, CTIINEN0

This register maps trigger inputs to channels in the cross trigger system. The CTIINEN registers are bit maps that allow the trigger input to be mapped to any channel output, including none (`0x0`) and all (`0xF`). There is one register per trigger input, so it is possible to map any combination of trigger inputs to any channel outputs.

The CTIINEN0 register characteristics are:

Attributes

Offset	`0x0020`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 9-485 CTIINEN0 register bit assignments

The following table shows the bit assignments.

Table 9-502 CTIINEN0 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGINEN	Trigger input to channel mapping. **0** Input trigger 0 events are ignored by the corresponding channel. **1** When an event is received on ctitrigin[0], generate an event on the channel corresponding to this bit.

CTI Trigger 1 to Channel Enable register, CTIINEN1

This register maps trigger inputs to channels in the cross trigger system. The CTIINEN registers are bit maps that allow the trigger input to be mapped to any channel output, including none (`0x0`) and all (`0xF`). There is one register per trigger input, so it is possible to map any combination of trigger inputs to any channel outputs.

The CTIINEN1 register characteristics are:

Attributes

Offset	`0x0024`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 9-486 CTIINEN1 register bit assignments

The following table shows the bit assignments.

Table 9-503 CTIINEN1 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGINEN	Trigger input to channel mapping. **0** Input trigger 1 events are ignored by the corresponding channel. **1** When an event is received on ctitrigin[1], generate an event on the channel corresponding to this bit.

CTI Trigger 2 to Channel Enable register, CTIINEN2

This register maps trigger inputs to channels in the cross trigger system. The CTIINEN registers are bit maps that allow the trigger input to be mapped to any channel output, including none (`0x0`) and all (`0xF`). There is one register per trigger input, so it is possible to map any combination of trigger inputs to any channel outputs.

The CTIINEN2 register characteristics are:

Attributes

Offset	`0x0028`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 9-487 CTIINEN2 register bit assignments

The following table shows the bit assignments.

Table 9-504 CTIINEN2 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGINEN	Trigger input to channel mapping. **0** Input trigger 2 events are ignored by the corresponding channel. **1** When an event is received on ctitrigin[2], generate an event on the channel corresponding to this bit.

CTI Trigger 31 to Channel Enable register, CTIINEN31

This register maps trigger inputs to channels in the cross trigger system. The CTIINEN registers are bit maps that allow the trigger input to be mapped to any channel output, including none (`0x0`) and all (`0xF`). There is one register per trigger input, so it is possible to map any combination of trigger inputs to any channel outputs.

The CTIINEN31 register characteristics are:

Attributes

Offset	`0x009c`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

Figure 9-488 CTIINEN31 register bit assignments

The following table shows the bit assignments.

Table 9-505 CTIINEN31 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGINEN	Trigger input to channel mapping. **0** Input trigger 31 events are ignored by the corresponding channel. **1** When an event is received on ctitrigin[31], generate an event on the channel corresponding to this bit.

CTI Channel to Trigger 0 Enable register, CTIOUTEN0

This register maps channels in the cross trigger system to trigger outputs. The CTIOUTEN registers are bit maps that allow any channel input to be mapped to the trigger output, including none (`0x0`) and all (`0xF`). There is one register per trigger output so it is possible to map any channel input to any combination of trigger outputs.

The CTIOUTEN0 register characteristics are:

Attributes

Offset	`0x00a0`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

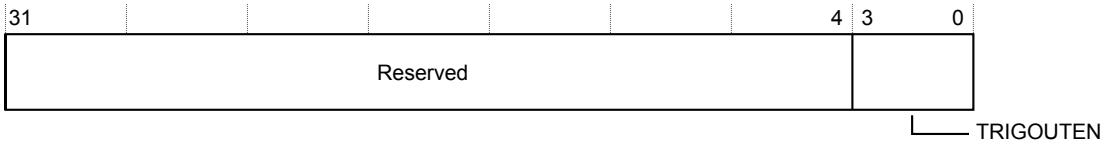

Figure 9-489 CTIOUTEN0 register bit assignments

The following table shows the bit assignments.

Table 9-506 CTIOUTEN0 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGOUTEN	Channel to trigger output mapping. **0** The corresponding channel is ignored by the output trigger0. **1** When an event occurs on the channel corresponding to this bit, generate an event on ctitrigout[0].

CTI Channel to Trigger 1 Enable register, CTIOUTEN1

This register maps channels in the cross trigger system to trigger outputs. The CTIOUTEN registers are bit maps that allow any channel input to be mapped to the trigger output, including none (`0x0`) and all (`0xF`). There is one register per trigger output so it is possible to map any channel input to any combination of trigger outputs.

The CTIOUTEN1 register characteristics are:

Attributes

Offset	`0x00a4`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

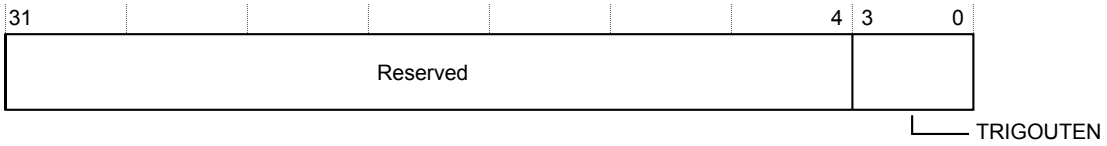

Figure 9-490 CTIOUTEN1 register bit assignments

The following table shows the bit assignments.

Table 9-507 CTIOUTEN1 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGOUTEN	Channel to trigger output mapping. **0** The corresponding channel is ignored by the output trigger1. **1** When an event occurs on the channel corresponding to this bit, generate an event on ctitrigout[1].

CTI Channel to Trigger 2 Enable register, CTIOUTEN2

This register maps channels in the cross trigger system to trigger outputs. The CTIOUTEN registers are bit maps that allow any channel input to be mapped to the trigger output, including none (`0x0`) and all (`0xF`). There is one register per trigger output so it is possible to map any channel input to any combination of trigger outputs.

The CTIOUTEN2 register characteristics are:

Attributes

Offset	`0x00a8`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

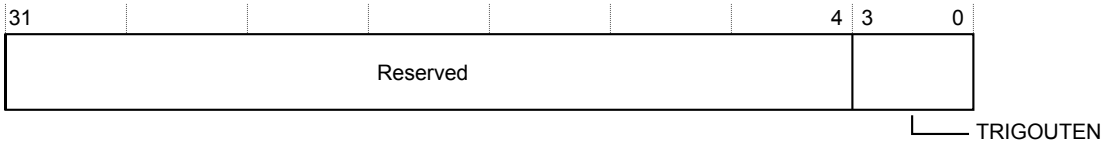

Figure 9-491 CTIOUTEN2 register bit assignments

The following table shows the bit assignments.

Table 9-508 CTIOUTEN2 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGOUTEN	Channel to trigger output mapping. **0** The corresponding channel is ignored by the output trigger2. **1** When an event occurs on the channel corresponding to this bit, generate an event on ctitrigout[2].

CTI Channel to Trigger 31 Enable register, CTIOUTEN31

This register maps channels in the cross trigger system to trigger outputs. The CTIOUTEN registers are bit maps that allow any channel input to be mapped to the trigger output, including none (`0x0`) and all (`0xF`). There is one register per trigger output so it is possible to map any channel input to any combination of trigger outputs.

The CTIOUTEN31 register characteristics are:

Attributes

Offset	`0x011c`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

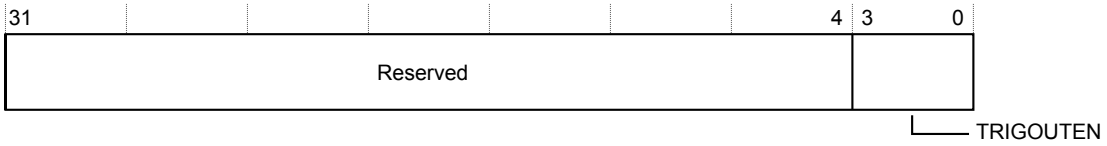

Figure 9-492 CTIOUTEN31 register bit assignments

The following table shows the bit assignments.

Table 9-509 CTIOUTEN31 register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	TRIGOUTEN	Channel to trigger output mapping. **0** The corresponding channel is ignored by the output trigger31. **1** When an event occurs on the channel corresponding to this bit, generate an event on ctitrigout[31].

CTI Trigger Input Status register, CTITRIGINSTATUS

Trigger input status. If the ctitrigin input is driven by a source that generates single cycle pulses, this register is generally read as 0.

The CTITRIGINSTATUS register characteristics are:

Attributes

Offset `0x0130`
Type Read-only
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

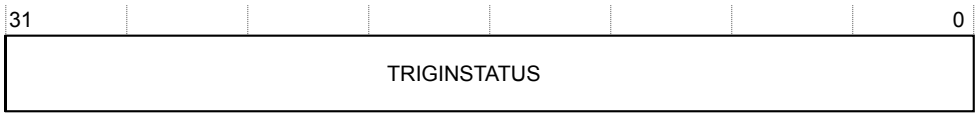

Figure 9-493 CTITRIGINSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-510 CTITRIGINSTATUS register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	TRIGINSTATUS	Trigger input status. **0** One bit per trigger input. 0 means that the input is LOW. **1** One bit per trigger input. 1 means that the input is HIGH.

CTI Trigger Output Status register, CTITRIGOUTSTATUS

Trigger output status. The register only has meaning if the trigger source drives static levels, rather than pulses.

The CTITRIGOUTSTATUS register characteristics are:

Attributes

Offset `0x0134`
Type Read-only
Reset `0x--------`
Width 32

The following figure shows the bit assignments.

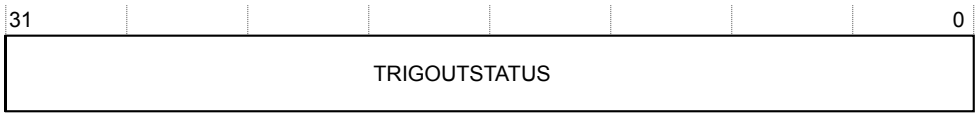

Figure 9-494 CTITRIGOUTSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-511 CTITRIGOUTSTATUS register bit assignments

Bits	Reset value	Name	Function
[31:0]	UNKNOWN	TRIGOUTSTATUS	Trigger output status. **0** One bit per trigger output. 0 means that the output is LOW. **1** One bit per trigger output. 1 means that the output is HIGH.

CTI Channel Input Status register, CTICHINSTATUS

Channel input status. If the channel input is driven by a source that generates single cycle pulses, this register is generally read as 0.

The CTICHINSTATUS register characteristics are:

Attributes

Offset	`0x0138`
Type	Read-only
Reset	`0x0000000-`
Width	32

The following figure shows the bit assignments.

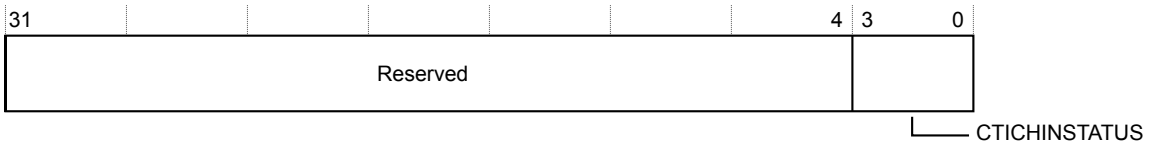

Figure 9-495 CTICHINSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-512 CTICHINSTATUS register bit assignments

Bits	Reset value	Name	Function
[3:0]	UNKNOWN	CTICHINSTATUS	Channel input status. **0** One bit per channel input. 0 means that the input is LOW. **1** One bit per channel input. 1 means that the input is HIGH.

CTI Channel Output Status register, CTICHOUTSTATUS

Channel output status. The register only has meaning if the trigger source drives static levels, rather than pulses.

The CTICHOUTSTATUS register characteristics are:

Attributes

Offset	0x013c
Type	Read-only
Reset	0x0000000-
Width	32

The following figure shows the bit assignments.

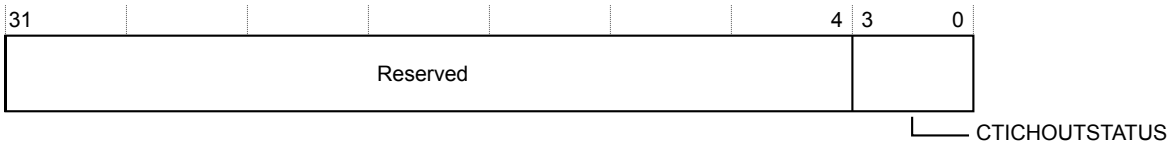

Figure 9-496 CTICHOUTSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-513 CTICHOUTSTATUS register bit assignments

Bits	Reset value	Name	Function
[3:0]	UNKNOWN	CTICHOUTSTATUS	Channel output status. **0** One bit per channel output. 0 means that the output is LOW. **1** One bit per channel output. 1 means that the output is HIGH.

Enable CTI Channel Gate register, CTIGATE

Channel output gate.

The CTIGATE register characteristics are:

Attributes

Offset	`0x0140`
Type	Read-write
Reset	`0x0000000f`
Width	32

The following figure shows the bit assignments.

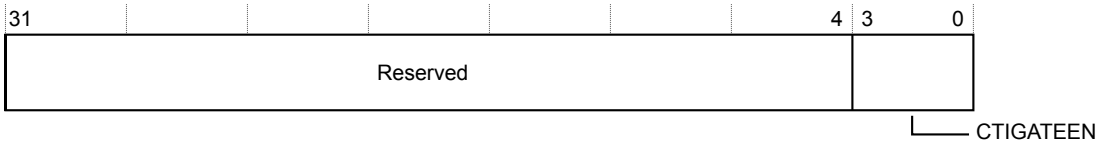

Figure 9-497 CTIGATE register bit assignments

The following table shows the bit assignments.

Table 9-514 CTIGATE register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	CTIGATEEN	Enables the propagation of channel events out of the CTI, one bit per channel. **0** Disable a channel from propagating. **1** Enable channel propagation.

External Multiplexer Control register, ASICCTRL

I/O port control. Controls the GPIO **asicctrl** output bus.

The ASICCTRL register characteristics are:

Attributes

Offset 0x0144
Type Read-write
Reset 0x00000000
Width 32

The following figure shows the bit assignments.

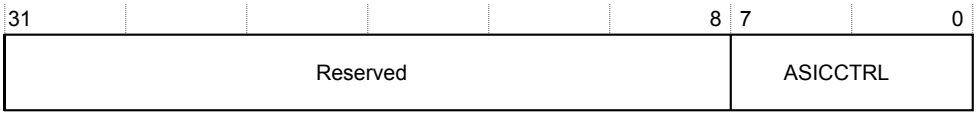

Figure 9-498 ASICCTRL register bit assignments

The following table shows the bit assignments.

Table 9-515 ASICCTRL register bit assignments

Bits	Reset value	Name	Function
[7:0]	0b00000000	ASICCTRL	Set and clear external output signal. **0** Clear output bit to 0. **1** Set output bit to 1.

Integration Test Channel Output register, ITCHOUT

Integration test mode register, used to generate channel events. Writing to the register creates a single pulse on the output. ITCHOUT is self-clearing.

The ITCHOUT register characteristics are:

Attributes

Offset	`0x0ee4`
Type	Read-write
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

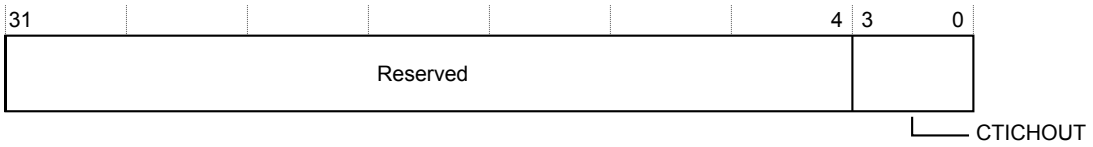

Figure 9-499 ITCHOUT register bit assignments

The following table shows the bit assignments.

Table 9-516 ITCHOUT register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	CTICHOUT	Pulses the channel outputs. **0** No effect. **1** Pulse channel event for one **clk** cycle.

Integration Test Trigger Output register, ITTRIGOUT

Integration test to generate trigger events.

The ITTRIGOUT register characteristics are:

Attributes

Offset `0x0ee8`
Type Read-write
Reset `0x00000000`
Width 32

The following figure shows the bit assignments.

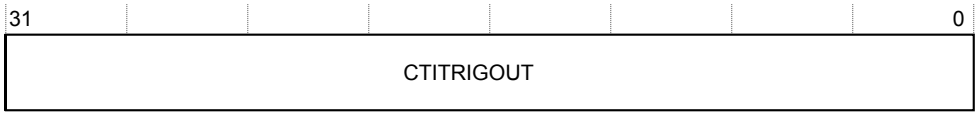

Figure 9-500 ITTRIGOUT register bit assignments

The following table shows the bit assignments.

Table 9-517 ITTRIGOUT register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	CTITRIGOUT	Set/clear trigger output signal. Reads return the value in the register. Writes: **0** Clears the trigger output. **1** Sets the trigger output.

Integration Test Channel Input register, ITCHIN

Integration test to view channel events. The integration test register includes a latch that is set when a pulse is received on a channel input. When read, a register bit reads as 1 if the channel has received a pulse since it was last read. The act of reading the register automatically clears the 1 to a 0. When performing integration testing it is therefore important to coordinate the setting of event latches and reading/clearing them.

The ITCHIN register characteristics are:

Attributes

Offset	`0x0ef4`
Type	Read-only
Reset	`0x00000000`
Width	32

The following figure shows the bit assignments.

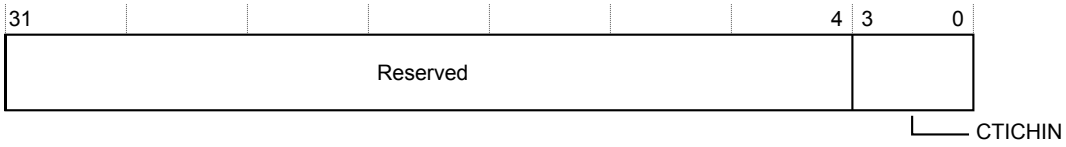

Figure 9-501 ITCHIN register bit assignments

The following table shows the bit assignments.

Table 9-518 ITCHIN register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	CTICHIN	Reads the latched value of the channel inputs.

Integration Test Trigger Input register, ITTRIGIN

Integration test to view trigger events. The integration test register includes a latch that is set when a pulse is received on a trigger input. When read, a register bit reads as 1 if the trigger input has received a pulse since it was last read. The act of reading the register automatically clears the 1 to a 0. When performing integration testing it is therefore important to coordinate the setting of event latches and reading/clearing them.

The ITTRIGIN register characteristics are:

Attributes

Offset `0x0ef8`

Type Read-only

Reset `0x00000000`

Width 32

The following figure shows the bit assignments.

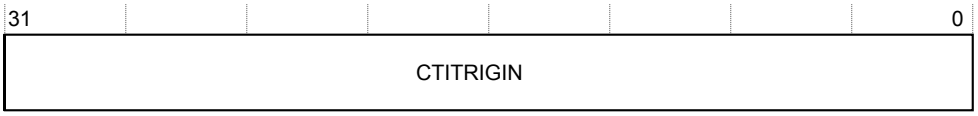

Figure 9-502 ITTRIGIN register bit assignments

The following table shows the bit assignments.

Table 9-519 ITTRIGIN register bit assignments

Bits	Reset value	Name	Function
[31:0]	`0x0`	CTITRIGIN	Reads the latched value of the trigger inputs.

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.

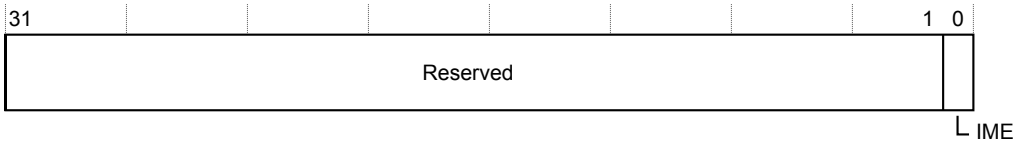

Figure 9-503 ITCTRL register bit assignments

The following table shows the bit assignments.

Table 9-520 ITCTRL register bit assignments

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

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 `0x0000000f`

Width 32

The following figure shows the bit assignments.

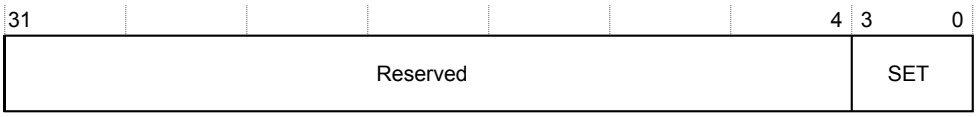

Figure 9-504 CLAIMSET register bit assignments

The following table shows the bit assignments.

Table 9-521 CLAIMSET register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b1111`	SET	A bit programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

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.

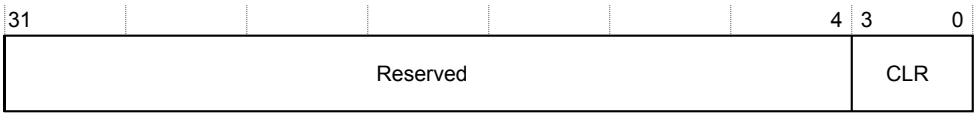

Figure 9-505 CLAIMCLR register bit assignments

The following table shows the bit assignments.

Table 9-522 CLAIMCLR register bit assignments

Bits	Reset value	Name	Function
[3:0]	`0b0000`	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.

Device Affinity register 0, DEVAFF0

Enables a debugger to determine if two components have an affinity with each other.

The DEVAFF0 register characteristics are:

Attributes

Offset `0x0fa8`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

Figure 9-506 DEVAFF0 register bit assignments

The following table shows the bit assignments.

Table 9-523 DEVAFF0 register bit assignments

Bits	Reset value	Name	Function
[31:0]	IMPLEMENTATION_DEFINED	DEVAFF0	Lower 32-bits of DEVAFF. The value is set by the **devaff[31:0]** tie-off inputs.

Device Affinity register 1, DEVAFF1

Enables a debugger to determine if two components have an affinity with each other.

The DEVAFF1 register characteristics are:

Attributes

Offset `0x0fac`

Type Read-only

Reset `0x--------`

Width 32

The following figure shows the bit assignments.

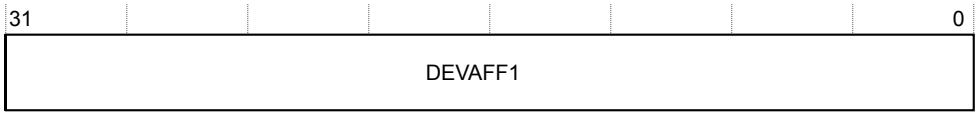

Figure 9-507 DEVAFF1 register bit assignments

The following table shows the bit assignments.

Table 9-524 DEVAFF1 register bit assignments

Bits	Reset value	Name	Function
[31:0]	IMPLEMENTATION_DEFINED	DEVAFF1	Upper 32-bits of DEVAFF. The value is set by the **devaff[63:32]** tie-off inputs.

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	`0x0000000-`
Width	32

The following figure shows the bit assignments.

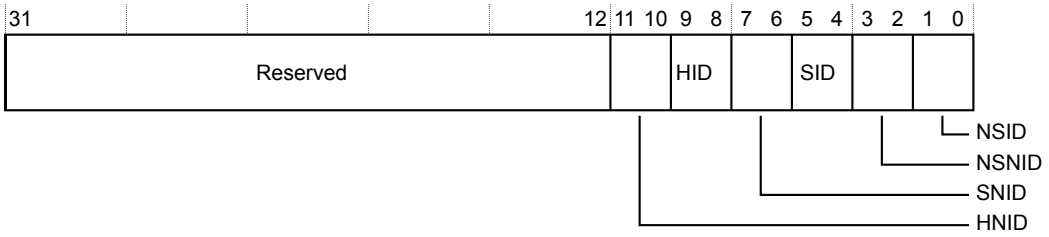

Figure 9-508 AUTHSTATUS register bit assignments

The following table shows the bit assignments.

Table 9-525 AUTHSTATUS register bit assignments

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]	`0b00`	SNID	Secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.
[5:4]	`0b00`	SID	Secure invasive debug. `0x0` Functionality not implemented or controlled elsewhere. `0x1` Reserved. `0x2` Functionality disabled. `0x3` Functionality enabled.

Table 9-525 AUTHSTATUS register bit assignments (continued)

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

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	`0x47701a14`
Width	32

The following figure shows the bit assignments.

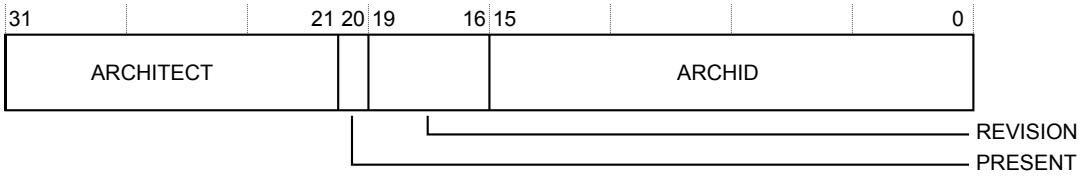

Figure 9-509 DEVARCH register bit assignments

The following table shows the bit assignments.

Table 9-526 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]	`0x1a14`	ARCHID	Architecture ID. Returns `0x1a14`, identifying Cross Trigger Interface architecture v2.

Device Configuration Register, DEVID

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

The DEVID register characteristics are:

Attributes

Offset 0x0fc8
Type Read-only
Reset 0x01040100
Width 32

The following figure shows the bit assignments.

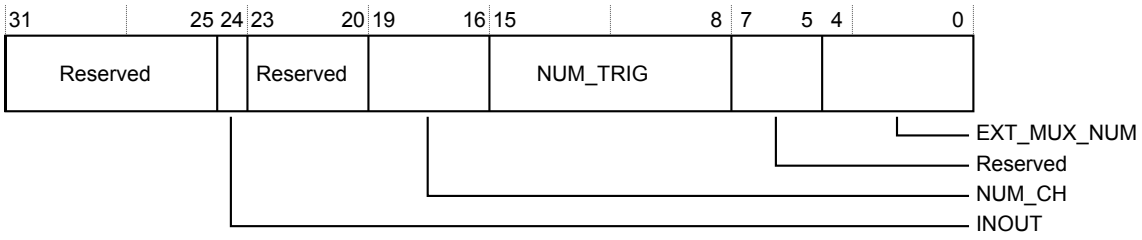

Figure 9-510 DEVID register bit assignments

The following table shows the bit assignments.

Table 9-527 DEVID register bit assignments

Bits	Reset value	Name	Function
[24]	0b1	INOUT	Indicates channel inputs are also masked by the CTIGATE register. Always 1.
[19:16]	0b0100	NUM_CH	The number of channels. Always 4.
[15:8]	0b00000001	NUM_TRIG	Indicates the maximum number of triggers - the maximum of the two parameters, NUM_EVENT_SLAVESand NUM_EVENT_MASTERS.
[4:0]	0b00000	EXT_MUX_NUM	Indicates the value of the EXT_MUX_NUMparameter, which determines if there is any external multiplexing on the trigger inputs and outputs. 0 indicates no multiplexing.

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 `0x00000014`
Width 32

The following figure shows the bit assignments.

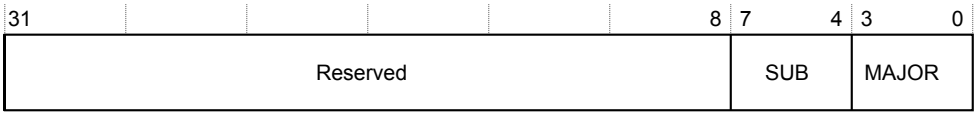

Figure 9-511 DEVTYPE register bit assignments

The following table shows the bit assignments.

Table 9-528 DEVTYPE register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0001`	SUB	Minor classification. Returns `0x1`, indicating this component is a Trigger-Matrix.
[3:0]	`0b0100`	MAJOR	Major classification. Returns `0x4`, indicating this component performs Debug Control.

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.

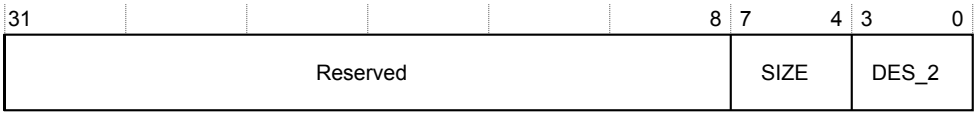

Figure 9-512 PIDR4 register bit assignments

The following table shows the bit assignments.

Table 9-529 PIDR4 register bit assignments

Bits	Reset value	Name	Function
[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.

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.

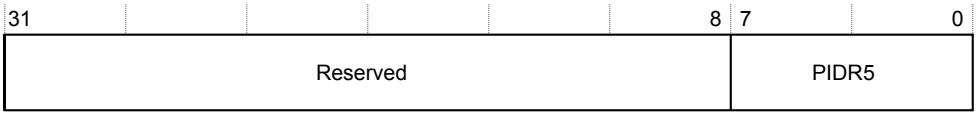

Figure 9-513 PIDR5 register bit assignments

The following table shows the bit assignments.

Table 9-530 PIDR5 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR5	Reserved.

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.

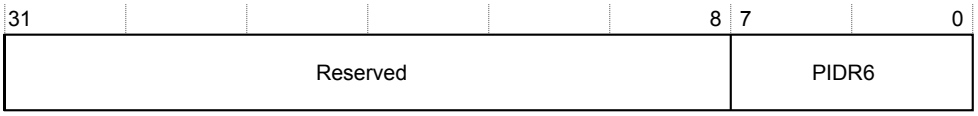

Figure 9-514 PIDR6 register bit assignments

The following table shows the bit assignments.

Table 9-531 PIDR6 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR6	Reserved.

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.

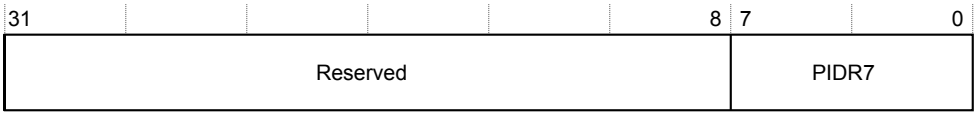

Figure 9-515 PIDR7 register bit assignments

The following table shows the bit assignments.

Table 9-532 PIDR7 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000000`	PIDR7	Reserved.

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 `0x000000ed`

Width 32

The following figure shows the bit assignments.

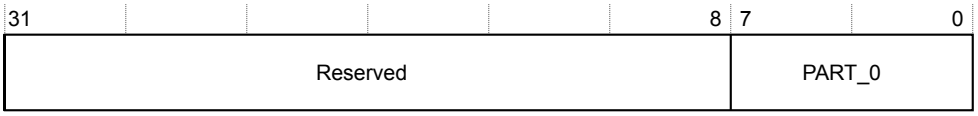

Figure 9-516 PIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-533 PIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b11101101`	PART_0	Part number, bits[7:0]. This is selected by the designer of the component.

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.

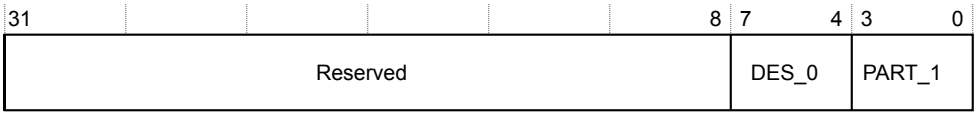

Figure 9-517 PIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-534 PIDR1 register bit assignments

Bits	Reset value	Name	Function
[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]. This is selected by the designer of the component.

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	`0x0000000b`
Width	32

The following figure shows the bit assignments.

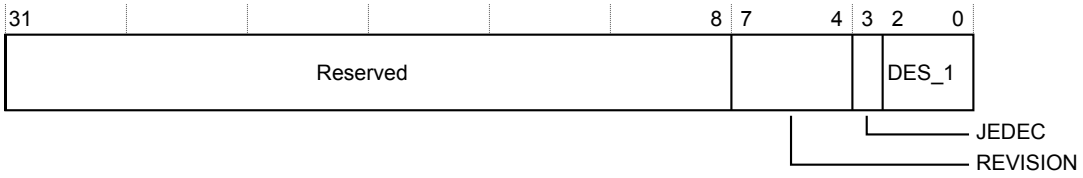

Figure 9-518 PIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-535 PIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b0000`	REVISION	Revision. It is an incremental value starting at `0x0` for the first design of a component. See the css600 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.

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.

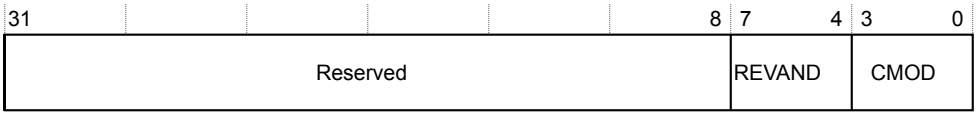

Figure 9-519 PIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-536 PIDR3 register bit assignments

Bits	Reset value	Name	Function
[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`.

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.

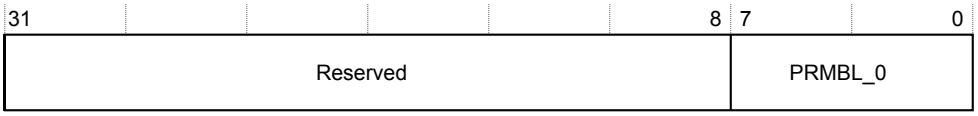

Figure 9-520 CIDR0 register bit assignments

The following table shows the bit assignments.

Table 9-537 CIDR0 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00001101`	PRMBL_0	Preamble. Returns `0x0D`.

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.

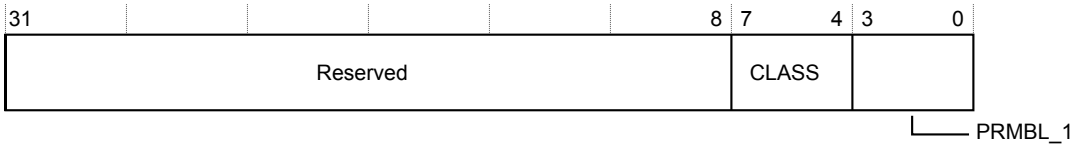

Figure 9-521 CIDR1 register bit assignments

The following table shows the bit assignments.

Table 9-538 CIDR1 register bit assignments

Bits	Reset value	Name	Function
[7:4]	`0b1001`	CLASS	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	`0b0000`	PRMBL_1	Preamble. Returns `0x0`.

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.

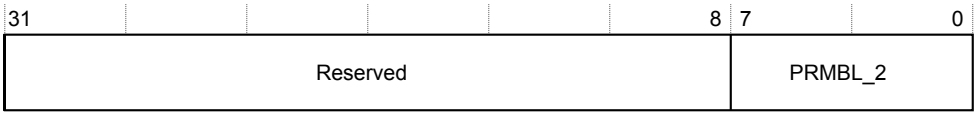

Figure 9-522 CIDR2 register bit assignments

The following table shows the bit assignments.

Table 9-539 CIDR2 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b00000101`	PRMBL_2	Preamble. Returns `0x05`.

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.

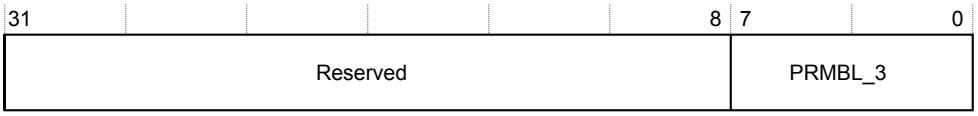

Figure 9-523 CIDR3 register bit assignments

The following table shows the bit assignments.

Table 9-540 CIDR3 register bit assignments

Bits	Reset value	Name	Function
[7:0]	`0b10110001`	PRMBL_3	Preamble. Returns `0xB1`.

Appendix A
Revisions

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

It contains the following section:

- *A.1 Revisions* on page Appx-A-702.

A.1 Revisions

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

Table A-1 Issue 0000-00

Change	Location	Affects
First release	-	-

Table A-2 Differences between issue 0000-00 and issue 0000-01

Change	Location	Affects
Updated component list	*1.6 css600 component list* on page 1-18	All revisions
Added DAP components chapter	*Chapter 2 DAP components functional description* on page 2-21	
Added APB infrastructure components chapter	*Chapter 3 APB infrastructure components functional description* on page 3-37	
Added ATB infrastructure components chapter	*Chapter 4 ATB infrastructure components functional description* on page 4-45	
Added Timestamp components chapter	*Chapter 5 Timestamp components functional description* on page 5-65	
Added Embedded cross-trigger components chapter	*Chapter 6 Embedded Cross Trigger components functional description* on page 6-76	
Added Authentication components chapter	*Chapter 7 Authentication components functional description* on page 7-92	
Updated DP programmers model section	*9.2 css600_dp introduction* on page 9-102	

Table A-3 Differences between issue 0000-01 and issue 0100-00

Change	Location	Affects
Added Narrow Timestamp components	*Chapter 5 Timestamp components functional description* on page 5-65	r1p0
Added PIL components	*Chapter 8 Processor Integration Layer components* on page 8-96	r1p0
Component list has been updated	*1.6 css600 component list* on page 1-18	r1p0
DEVID1 added to the TMC programmers models	*9.12.1 Register summary* on page 9-424	r1p0
PIDR2 has been updated in the TMC programmers models	*9.12.1 Register summary* on page 9-424	r1p0