CoreSight™ Trace Memory Controller

Revision: r0p1

Technical Reference Manual

ARM®

Copyright © 2010 ARM. All rights reserved.
ARM DDI 0461B (ID010111)

CoreSight Trace Memory Controller
Technical Reference Manual

Copyright © 2010 ARM. All rights reserved.

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
22 June 2010	A	Non-Confidential	First release for r0p0
10 December 2010	B	Non-Confidential	First release for r0p1

Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Where the term ARM is used it means “ARM or any of its subsidiaries as appropriate”.

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.

Product Status

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

Web Address

`http://www.arm.com`

Contents
CoreSight Trace Memory Controller Technical Reference Manual

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Preface

This preface introduces the *CoreSight Trace Memory Controller Technical Reference Manual*. It contains the following sections:

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

About this book

This book is for CoreSight *Trace Memory Controller* (TMC).

Product revision status

The r*n*p*n* identifier indicates the revision status of the product described in this book, where:

r*n* Identifies the major revision of the product.

p*n* Identifies the minor revision or modification status of the product.

Intended audience

This book is written for system designers, system integrators, and programmers who are designing or programming a *System-on-Chip* (SoC) that uses the TMC.

Using this book

This book is organized into the following chapters:

Chapter 1 *Introduction*

Read this for an introduction to the TMC and its features.

Chapter 2 *Functional Description*

Read this for a description of the major interfaces and the functional operation of the TMC.

Chapter 3 *Programmers Model*

Read this for a description of the memory maps and registers.

Appendix A *Signal Descriptions*

Read this for a description of the input and output signals.

Appendix B *Revisions*

Read this for a description of the technical differences between consecutive revisions of this book.

Glossary Read the glossary for definitions of terms used in this book.

Conventions

Conventions that this book can use are described in:

- *Typographical*
- *Timing diagrams* on page vii
- *Signals* on page vii.

Typographical

The typographical conventions are:

italic Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.

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

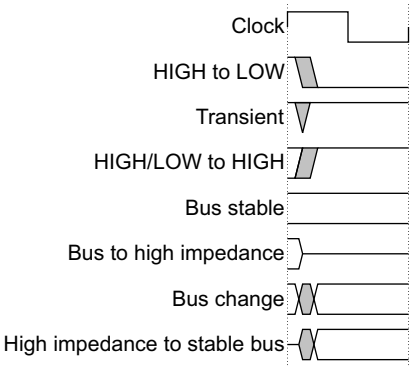
Copyright © 2010 ARM. All rights reserved.
Non-Confidential

`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 >	Enclose replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>`

Timing diagrams

The figure named *Key to timing diagram conventions* 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.



Key to timing diagram conventions

Timing diagrams sometimes show single-bit signals as HIGH and LOW at the same time and they look similar to the bus change shown in *Key to timing diagram conventions*. If a timing diagram shows a single-bit signal in this way then its value does not affect the accompanying description.

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.

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

Additional reading

This section lists publications by ARM and by third parties.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

See Infocenter, http://infocenter.arm.com, for access to ARM documentation.

ARM publications

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

- *CoreSight Architecture Specification v1.0* (ARM IHI 0029)
- *CoreSight Components Technical Reference Manual* (ARM DDI 0314)
- *High Speed Serial Trace Port Architecture Specification* (PR106-PRDC-006159)
- *CoreSight Trace Memory Controller (TMC) Implementation and Integration Manual* (PR430-PRDC-011743)
- *AMBA AXI Protocol Specification* (ARM IHI 0022)
- *AMBA 3 APB Protocol Specification* (ARM 0024)
- *AMBA 3 ATB Protocol Specification* (ARM IHI 0032A).

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Feedback

ARM welcomes feedback on this product and its documentation.

Feedback on this product

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

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

Feedback on content

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

- the title
- the number, ARM DDI 0461B
- the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Chapter 1
Introduction

This chapter introduces the *Trace Memory Controller (TMC)*. It contains the following sections:

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

1.1 About the TMC

The *Trace Memory Controller* (TMC) is designed as a successor to the *CoreSight Embedded Trace Buffer* (CoreSight ETB) that enables you to capture trace using:

- the debug interface such as 2-pin serial wire debug
- the system memory such as a dynamic *Random Access Memory* (RAM)
- the high-speed links that already exist in the *System-on-Chip* (SoC) peripheral.

The TMC can also operate as a *First In First Out* (FIFO) in the system. This reduces the trace overflow and trace port size by averaging out the trace bandwidth.

The following sections describe:

- *Configuration options*
- *Operational modes* on page 1-3.

1.1.1 Configuration options

The TMC has the following major configurations that are fixed when the design is integrated into a system:

***Embedded Trace Buffer* (ETB)**

Enables trace to be stored in a dedicated SRAM, used as a Circular Buffer. This configuration is similar to the CoreSight ETB.

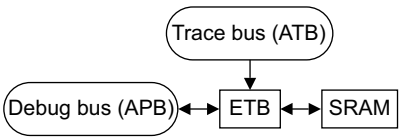


Figure 1-1 ETB configuration

***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 this configuration is a superset of the functionality of the ETB configuration.

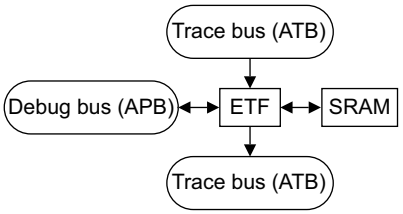


Figure 1-2 ETF configuration

***Embedded Trace Router* (ETR)**

Enables trace to be routed over an AXI bus to system memory or to any other AXI slave.

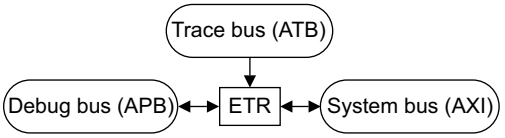


Figure 1-3 ETR configuration

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

1.1.2 Operational modes

You can program the TMC to capture trace in one of the following modes:

Circular Buffer mode

This is the mode supported by the CoreSight ETB, and it is available in all configurations. The component captures trace, using its storage as a circular buffer, overwriting old trace once the buffer is full. No trace can be read until trace capture has stopped. In this mode, trace capture can automatically stop after receiving a trigger signal.

Hardware FIFO mode

This mode is available only in the ETF configuration. The component uses its storage as a FIFO and acts as a link between a trace source, and a trace sink. No trace is lost or overwritten, and back pressure is applied through ATB to the trace source when the FIFO becomes full. Most trace sources eventually overflow when subject to back-pressure for a long time, but it is always the trace sources that lose trace, not the ETF. This mode enables peaks in trace bandwidth requirement to be smoothed, and reduces the requirement to exert back-pressure, 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
- trace that is output over a subsequent ETR can cope with memory system latency caused by higher priority masters on the interconnect, without the loss of trace.

Software FIFO mode

This mode is available in all configurations. In this mode, the component functions as a FIFO, where data is read out over an APB interface, usually by an external debugger. This provides a low-speed off-chip communication channel for retrieving trace data, reusing the existing JTAG or *Serial Wire Debug* (SWD) connection.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

1.2 Example systems with different configurations

This section uses example systems to demonstrate the different TMC configurations that you can use as an individual TMC implementation or as a combination of different TMC configuration:

- *CoreSight ETB and TPIU*
- *ETF and TPIU*
- *ETB only* on page 1-5
- *ETR to SDRAM* on page 1-5
- *ETF, ETR, and TPIU* on page 1-6
- *ETR to alternative interface* on page 1-7
- *Local ETF* on page 1-8.

These given systems are built using the TMC and other CoreSight components.

1.2.1 CoreSight ETB and TPIU

Figure 1-4 shows the usual system used when integrating the CoreSight ETB and TPIU.

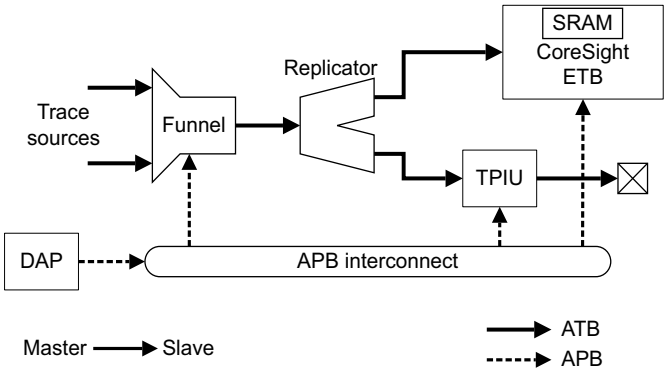


Figure 1-4 CoreSight ETB and TPIU

The system supports the following usage models:

- trace capture in dedicated SRAM when ETB is enabled
- trace capture in an off-chip capture device when TPIU is enabled.

1.2.2 ETF and TPIU

Figure 1-5 shows an example system that directly replaces the *CoreSight ETB and TPIU* when no other TMC functionality is required.

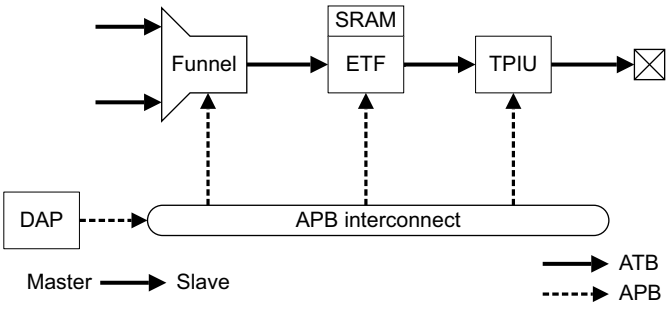


Figure 1-5 ETF and TPIU

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

The system supports the following usage models:

- Trace capture in dedicated SRAM within ETF programmed in Circular Buffer mode. This usage model is unchanged from *CoreSight ETB and TPIU* on page 1-4. However, when the trace capture has stopped, the trace can be drained through the trace port through the TPIU. This is considerably faster than downloading the trace through the *Debug Access Port* (DAP).
- Trace capture in an off-chip capture device, with on-chip buffering. The on-chip buffering, performed by the ETF, enables peaks in trace generation to be smoothed, and reduces the number of trace pins required.
- Low-speed off-chip trace capture through the DAP. In this model, the TPIU is disabled and the ETF is programmed to be in Software FIFO mode. Trace is downloaded by the DAP in real-time, and the ETF RAM is used to buffer the trace prior to download. This model is useful when the trace port is not available, for example when debugging a final product.

1.2.3 ETB only

Figure 1-6 shows an example system that is similar to *ETF and TPIU* on page 1-4.

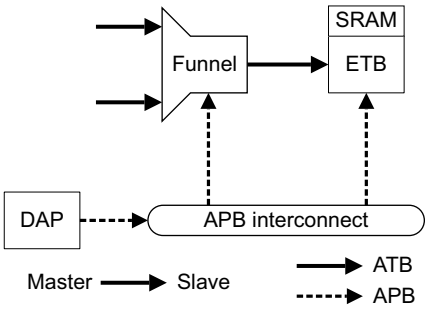


Figure 1-6 ETB only

However, in this system, it is not possible to implement a trace port. The system supports the following usage models:

- Trace capture in dedicated SRAM. When captured, the trace must be downloaded through the DAP.
- Low-speed off-chip trace capture. This usage model is the same as low-speed off-chip trace capture through the DAP as mentioned in *ETF and TPIU* on page 1-4.

1.2.4 ETR to SDRAM

Figure 1-7 on page 1-6 shows an example system that enables other forms of dedicated memory to be used for on-chip trace capture, where a *Dynamic Memory Controller* (DMC) controls the memory.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

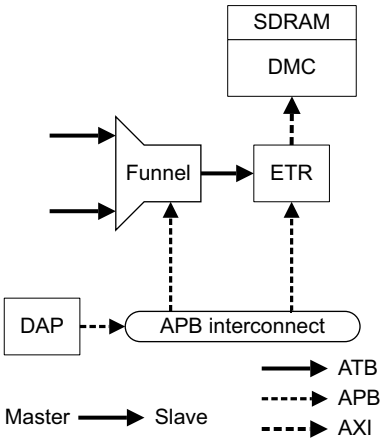


Figure 1-7 ETR to SDRAM

This system supports the following usage models:

- Trace capture in dedicated SDRAM.
- Low-speed off-chip trace capture through the DAP. The ETR can be used as an intermediate buffer in FIFO mode, and operates in the same way as the ETB or ETF in this mode.

By using a DMC, you can use other memory technologies with higher densities. However, the area savings must be balanced against the increased area of the combined ETR and DMC compared to an ETB. This system is therefore most applicable where a large dedicated trace RAM is required.

The ETR and DMC might be connected by a sparse interconnect instead of being directly connected.

The ETR supports a configurable write buffer depth, and programmable burst options for optimum efficiency of the DMC.

1.2.5 ETF, ETR, and TPIU

Figure 1-8 on page 1-7 shows an example system that combines multiple TMC configurations to support many debug usage models with various trade-offs between invasiveness, trace depth, and trace bandwidth. It is likely to be the most common system for high-end systems.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

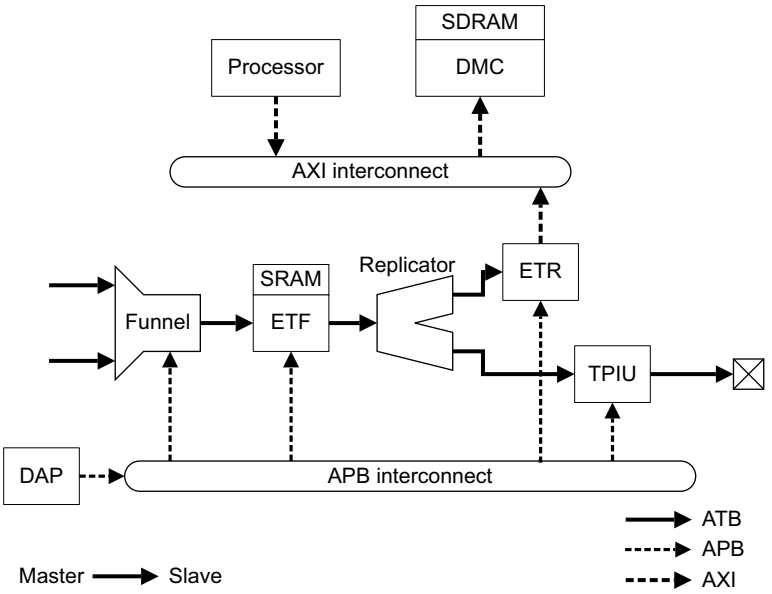


Figure 1-8 ETF, ETR, and TPIU

This system supports the following usage models:

- Trace capture in dedicated SRAM are stored in the ETF. When trace capture has stopped, it can be downloaded through the trace port. Trace capture is fully non-intrusive and high bandwidth, but of limited depth.
- Trace capture in an off-chip capture device with on-chip buffering. Trace capture is fully non-intrusive and its depth is limited only by the off-chip capture device, but bandwidth is limited by the trace port.
- Trace capture in system memory with intermediate buffering. The intermediate buffering enables the trace to be resilient to large delays on the interconnect. The ETR also supports programmable burst options for optimum efficiency of the interconnect and DMC. Trace capture can be intrusive to system performance, because less memory is available to the rest of the system, and because the bandwidth of the DMC must be shared between the ETR and regular system functions. However, the depth of trace capture available is likely to be much larger than available when storing in the ETF, and a trace port is not required.
- Low-speed off-chip trace capture through the DAP. For this usage model, the ETR and TPIU are disabled, and the ETF is programmed to be in Software FIFO mode. Trace capture is fully non-intrusive, and can be of high depth, but with very limited bandwidth.

Additional systems are possible where the ETR is used to write data to system memory without going through an intermediate ETF. However, more careful design of the interconnect is required in such systems, to ensure that large delays on the interconnect do not result in trace loss. A write buffer in the ETR can be configurable up to 32 entries and implemented in D-type registers. It can reduce the impact of interconnect delays. An intermediate ETF is recommended where greater buffering is required.

1.2.6 ETR to alternative interface

Figure 1-9 on page 1-8 shows an example system that enables the trace to be carried off-chip by an interface other than the CoreSight trace port.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

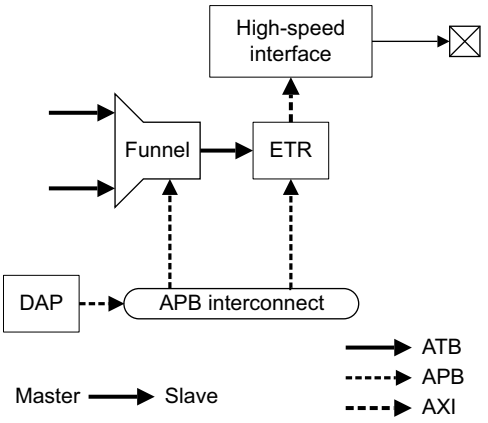


Figure 1-9 ETR to alternative interface

This system supports the following usage model:

- Trace capture in an off-chip device. The communications protocol is defined by the high-speed interface IP to which the ETR is connected.

You can configure the size of the write buffer in the ETR according to the requirements of the high-speed interface. If the high-speed interface accepts data at a steady rate, you can configure the ETR with a minimal write buffer to reduce the area.

1.2.7 Local ETF

Figure 1-10 shows an example system where ETBs and ETFs can be implemented locally to individual trace sources or groups of trace sources where the total bandwidth requirements of the system are very high.

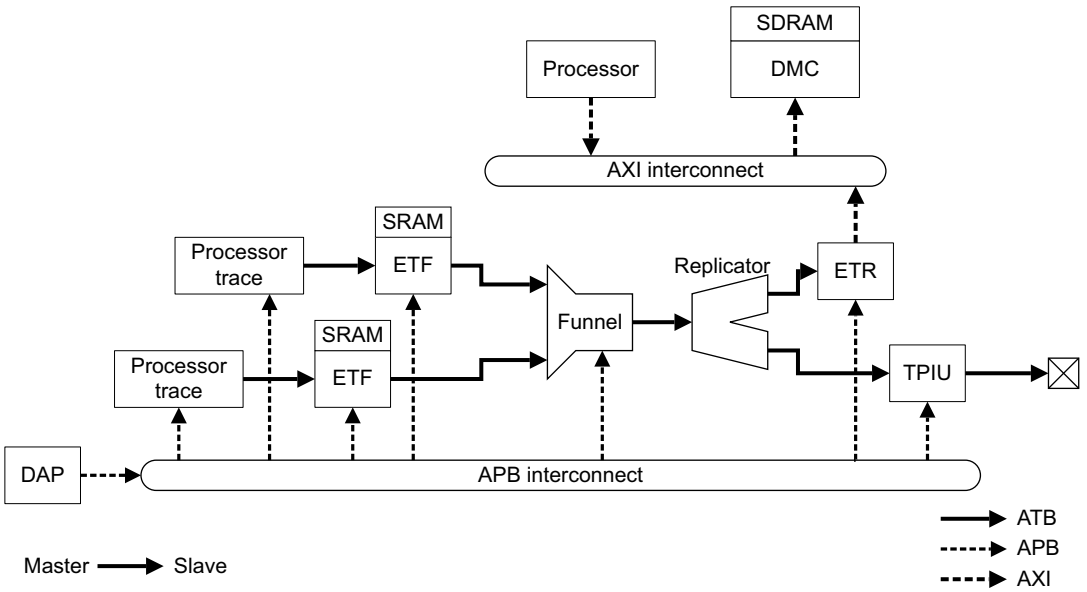


Figure 1-10 Local ETF

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

This system supports all of the usage models that *ETF, ETR, and TPIU* on page 1-6 support. This system has the following additional advantages:

- When trace is being captured in the ETFs in Circular Buffer mode, greater total trace bandwidth is available. If each ETF is connected to a number of trace sources, the maximum trace bandwidth for the whole system might exceed the available output of the funnel. Capturing trace locally avoids this bottleneck, and enables more trace sources to be enabled simultaneously.
- When trace is being captured in system memory through the ETR or off-chip by the TPIU, the system can accommodate bursts of trace data that exceed the available trace bandwidth on the output of the funnel, without causing overflows.

These benefits must be balanced against the following disadvantages:

- there is an increased area cost because of multiple ETFs
- the RAM associated with each ETF can only be used by the trace sources connected to it.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

1.3 Features of TMC

The TMC provides the following features:

- *Configurable parameters*
- *Backward-compatible*
- *Connections to the CTI*
- *Scatter-gather table* on page 1-11
- *Buffer drain over ATB* on page 1-11
- *Synchronization request* on page 1-11
- *AXI master interface* on page 1-11
- *Integration registers* on page 1-12.

1.3.1 Configurable parameters

The TMC is configurable with the following parameters:

ATBM_AXIM_CFG

Selects ETB, ETF, or ETR configuration.

MEM_SIZE

This parameter is only used in ETB and ETF configurations. Selects the SRAM memory size from 512 Bytes to 4GB, in powers of two.

WBUFFER_DEPTH

This parameter is only used in ETR configuration. The ETR incorporates an internal write buffer to overcome delays on the interconnect that can be configured to be from four to 32 beats deep.

ATB_DATA_WIDTH

Selects the width of the ATB interfaces, and the width of the full data path through the TMC. The ATB interfaces are always of the same width and can be configured to be 32 bits, 64 bits, and 128 bits wide.

In ETR configuration, the AXI master interface is always the same width as the ATB interfaces. In ETB and ETF configurations, the memory interface is always twice the width of the ATB interfaces.

1.3.2 Backward-compatible

The ETB configuration of the TMC with 32-bit ATB is largely backwards-compatible with the CoreSight ETB. However, some minor changes might be required, especially if non-standard usage models are employed.

See *Backwards compatibility* on page 2-21.

1.3.3 Connections to the CTI

All the CTI connections available in the CoreSight ETB are available in the TMC. See the *CoreSight Components Technical Reference Manual*.

You can route the following signals through the CTI to any of the trigger outputs in your Embedded Cross Trigger network:

FULL An output signal indicating when the Circular Buffer or FIFO is full, or within a programmable amount of being full.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

ACQCOMP

An output signal indicating when trace capture has stopped, usually following a trigger condition.

TRIGIN An input signal indicating when a trigger condition has occurred.

FLUSHIN An input signal indicating a flush request.

For example, you can program **ACQCOMP**, an output signal, to cause a processor interrupt when acquisition is complete. This can be useful for target-hosted debug tools. See *Event interfaces* on page 2-4.

1.3.4 Scatter-gather table

The ETR supports the use of a scatter-gather table that indicates the physical address regions to use. In most operating systems, it is not possible to guarantee the availability of memory with contiguous physical addresses after boot. To overcome this problem, the ETR supports the use of a scatter-gather table that indicates the location of physical memory blocks.

This scatter-gather table is setup by the operating system using a device driver, and is read automatically by the ETR. Entries in the table consist of the following formats:

- an entry indicating the base address of a 4KB block of memory to be used for trace storage.
- an entry indicating the address of a continuation table, in case the size of the scatter-gather table itself is too large to fit in a region of contiguous physical address space.
- an entry indicating the end of the table.

See *Scatter-gather* on page 2-16.

1.3.5 Buffer drain over ATB

The TMC in ETF configuration can drain the contents of its memory over the ATB interface for fast download to a debugger following the trace capture in Circular Buffer mode.

If the trace is captured in system memory, the TMC does not enable the trace to be downloaded over the trace port, because the ETR does not have an ATB master interface. However, the trace can be downloaded through the APB slave interface.

A separate trace component like *System Trace Macrocell* (STM) can also be used to dump the contents of memory over the trace port.

1.3.6 Synchronization request

The TMC can generate a periodic synchronization request signal, **SYNCREQ**, to coordinate the generation of synchronization packets by different trace sources capable of receiving such a signal, like the Cortex™ - A9 PTM. This ensures that the trace generated from all trace sources can be uncompressed when the start of trace is lost because of capture in a circular buffer, and when the trace from each source is being generated at a highly variable rate. See *Synchronisation request interface* on page 2-7.

1.3.7 AXI master interface

The AXI master interface supports bursts of up to 16 beats, depending on the configured depth of the write buffer. This maximizes the efficiency of the memory when writing to DRAM. See *AXI master interface* on page 2-4.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

1.3.8 Integration registers

The TMC supports a full set of integration registers, compatible with those available in the CoreSight ETB, to simplify the integration testing.

1.3.9 Clock gating

The TMC implements an automatic clock gate for power saving, so an external clock gate is not required. When the TMC is not in use, only the APB slave interface is clocked.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

1.4 Product revisions

This section describes the differences in functionality between product revisions:

r0p0 First release.

r0p0 - r0p1 This release modifies the revision field to `0x1`, see *Peripheral ID2 Register* on page 3-45.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Chapter 2
Functional Description

This chapter describes the interfaces, operation, clocking, and resets of the TMC. It contains the following sections:

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.1 Functional interfaces

The TMC has the following major interfaces:

- *Memory interface*
- *ATB interfaces* on page 2-3
- *CLK, PCLKENDBG, and RESETn* on page 2-3
- *AXI master interface* on page 2-4
- *Event interfaces* on page 2-4
- *APB slave interface* on page 2-4
- *AXI low-power interface* on page 2-4
- *Authentication interface* on page 2-7
- *Test* on page 2-7
- *Synchronisation request interface* on page 2-7.

2.1.1 Memory interface

In ETB and ETF configurations, the memory interface supports access to on-chip SRAM to store and retrieve trace data. The embedded SRAM is not instantiated inside the TMC, but it is connected through an external interface. The SRAM must:

- Be single ported.
- Have zero cycle latency. Data is available one cycle after the address is presented.
- Have zero bus turn around. This enables reads and writes to be interleaved without penalty. This is essential for Hardware FIFO mode to operate efficiently.
- Have twice the width of the ATB data width. This ensures full throughput in FIFO mode without requiring dual-ported memory.

Table 2-1 shows the operation of the memory interface signals.

Table 2-1 Memory interface signals

Signal	Description
CLK	The input clock for the design.
MEMCEN	Chip enable, active LOW.
MEMADDR	Address.
MEMWEN	Write enable, active LOW.
MEMD	Write data. The data is valid in the same cycle as the address.
MEMQ	Read data. The data is returned one cycle after the address.

Figure 2-1 on page 2-3 shows the operation of the memory interface during interleaved reads and writes. When Hardware FIFO mode is selected, reads and writes are interleaved. See Figure 2-1 on page 2-3.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

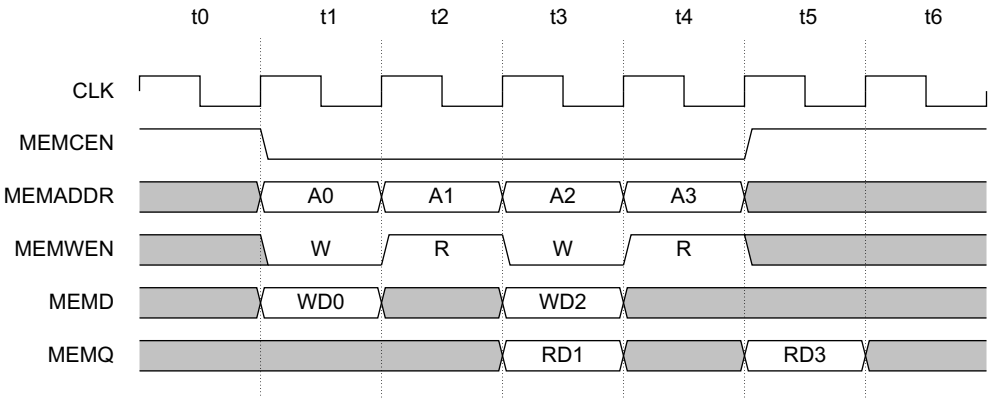


Figure 2-1 Interleaved memory interface operation

Figure 2-1 shows the following operations:

- a write to address A0 in cycle t1 where the data WD0 is also presented on **MEMD**
- a read from address A1 in cycle t2, and the data RD1 is returned on **MEMQ** in cycle t3
- a write to address A2 in cycle t3 where the data WD2 is also presented on **MEMD**
- a read from address A3 in cycle t4, and the data RD3 is returned on **MEMQ** in cycle t5.

The signals of the memory interface are fully registered.

2.1.2 ATB interfaces

An ATB slave interface is used to receive the trace data. It can be connected to any component with a standard ATB master interface. It can support a configurable data width of 32-bit, 64-bit, or 128-bit.

In the ETF configuration, an ATB master interface enables draining of data from the local SRAM. It can be connected to any component with a standard ATB slave interface. The configurable data width is equal to the ATB slave interface data width.

For more information about these interfaces, see the *AMBA 3 ATB Protocol Specification*.

2.1.3 CLK, PCLKENDBG, and RESETn

This section describes:

- *CLK*
- *PCLKENDBG*
- *RESETn* on page 2-4.

CLK

All configurations of the TMC use a single clock domain with input, **CLK**. See *Clocks and resets* on page A-2.

PCLKENDBG

PCLKENDBG enables the APB slave interface to connect to a slower APB master interface, with a clock synchronous to the TMC clock.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

RESETn

The TMC provides a single reset input, **RESETn**, that is active LOW. A TMC in an ETR configuration can be in a different power or reset domain to the AXI slave interface it is connected to.

For more information about **RESETn** and AXI low-power interface functionality, see *AXI low-power interface*.

2.1.4 AXI master interface

In ETR configuration, the AXI master interface takes the place of the memory interface used in ETB and ETF configurations.

The AXI master interface has the following properties:

- all transfers are the full width of the data bus
- only master ID 0 is used
- no locked transfers are issued
- no bursts are performed on the read interface, and the burst type is always indicated as a fixed-address burst of length one
- bursts of up to 16 beats can be performed on the write interface, and the burst type is always indicated as an incrementing burst.

The AXI master interface supports up to 32 outstanding write transactions. If an error response is returned at any time, it stops the operation until the debugger identifies and clears the error condition. See *Status Register* on page 3-6.

2.1.5 Event interfaces

The following event interfaces exist:

TRIGIN This input can cause a Trigger Event. See *Trigger, flush and stop events* on page 2-13.

FLUSHIN This input can cause a trace flush. See *Trigger, flush and stop events* on page 2-13.

FULL When the TMC is not in integration mode, this output indicates the value of the Full bit in the STS Register. See *Status Register* on page 3-6.

ACQCOMP When the TMC is not in integration mode, this output indicates the value of the FtEmpty bit in the STS Register. See *Status Register* on page 3-6.

You must connect these signals to a CoreSight *Cross Trigger Interface* (CTI).

2.1.6 APB slave interface

An APB slave interface is used for accessing the TMC registers.

2.1.7 AXI low-power interface

In an ETR configuration, an AXI low-power interface is provided to enable a reset controller or power controller to ensure that the AXI master interface is stable, with no outstanding operations, before a reset or power-down of the TMC.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

For example, an external debugger can request, through the DAP, that the debug logic is reset. Although it is unusual for a debug reset to be requested when the TMC is in use, it is possible for a debugger to cause the TMC to be reset at the same time that transactions are in progress on the AXI master interface, and this could cause an AXI protocol violation. You can guarantee that no AXI protocol violations are possible, by performing a handshake on an AXI low-power interface before asserting the debug reset.

Similarly, by using the AXI low-power interface, you can ensure that the AXI signals are stable before a reset is performed. This eliminates the risk of metastability caused by setup and hold time violations when the signals on the AXI master interface are asynchronously reset in the middle of a cycle. Figure 2-2 on page 2-6 shows the operation of the AXI low-power interface with **RESETn**.

The *AMBA AXI Protocol Specification defines the AXI low-power interface*. The signals are as follows:

CSYSREQ An input signal that is active LOW. It requests an AXI stall. When asserted:

- no more bursts are started on the AW or AR channels
- any bursts currently in progress complete, with beats continuing to be issued on the W channel.

CSYSACK An output signal that is active LOW. It indicates that the AXI master interface is stalled. When asserted:

- no read or write transactions are in progress
- no more activity occurs on the AXI master interface until **CSYSREQ** is driven HIGH.

CACTIVE An output signal that is active LOW. It indicates that a low-power request is accepted. It is always driven LOW, because a low-power request is never refused.

Figure 2-2 on page 2-6 shows the permitted operation of the AXI master interface, along with the reset.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

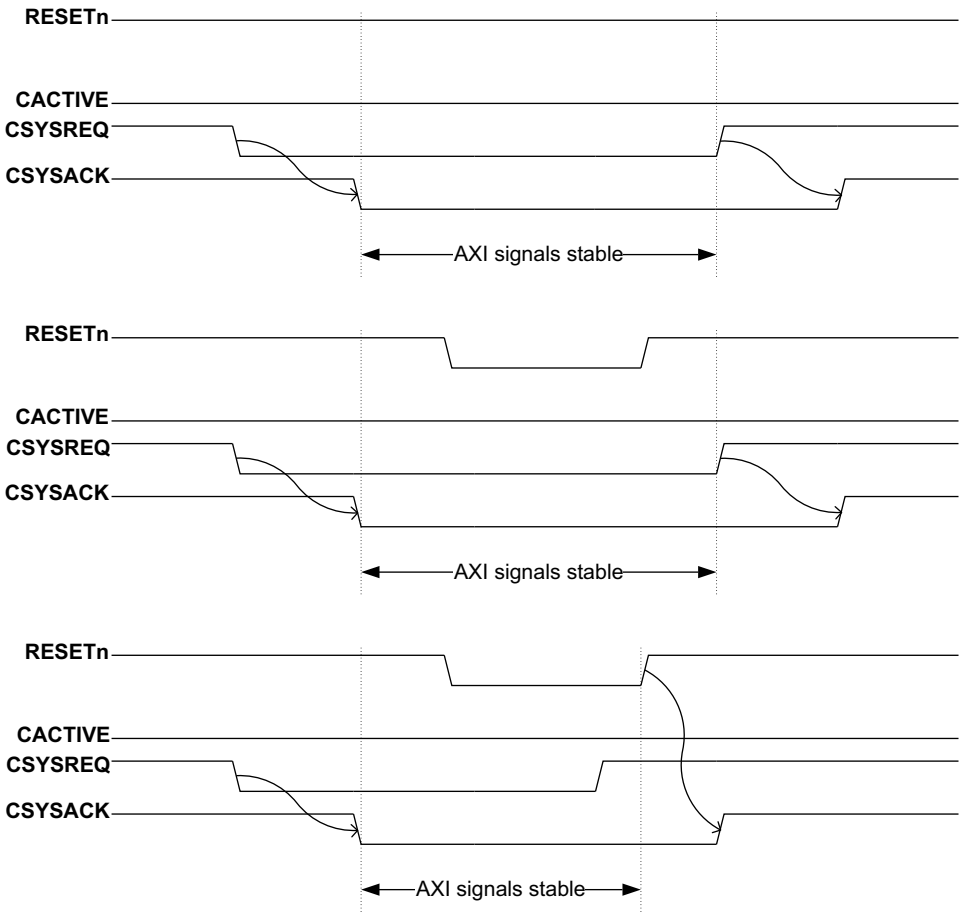


Figure 2-2 AXI low-power operation

In Figure 2-2:

- In the first example, **CSYSREQ** and **CSYSACK** follow a four-phase handshake sequence without a reset.
- In the second example, a reset is performed while the AXI signals are stable. **CSYSACK** is reset LOW, and therefore starts in the AXI stable state.
- In the third example, **CSYSREQ** is driven HIGH while **RESETn** is still asserted. **CSYSACK** goes HIGH when **RESETn** is deasserted.

CSYSREQ must be driven to maintain the four-phase handshake protocol. The AXI low-power interface only stalls the activity on the AXI low-power interface. For example:

- any APB accesses to the RRD or RWD registers stall until **CSYSREQ** is driven HIGH
- if the TMC is enabled, the ATB slave interface stalls when the write buffer becomes full, because the TMC can no longer write to memory.

Note

The AXI low-power interface does not indicate when all trace data has been drained from the TMC, for example, prior to removing the power from the TMC. If this is required, the TMC must be stopped and disabled in software.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.1.8 Authentication interface

In ETR configuration, the **DBGEN** and **SPIDEN** signals of the authentication interface affect the behavior of the TMC as follows:

- If **DBGEN** is LOW, no AXI accesses are possible. Any attempted AXI access behaves as if an immediate error response had been returned, and sets the Memory Error Status bit in the STS Register. No transactions are issued on the AXI master port.
- If **SPIDEN** is LOW, no secure AXI accesses are possible. Any attempted AXI access while ProtCtrlBit1 in the AXICTL Register is clear behaves as if an immediate error response had been returned, and sets the Memory Error Status bit in the STS register. No transactions are issued on the AXI master port.

Relationship to authentication interface signals

The **NIDEN** and **SPNIDEN** signals are not implemented, because the TMC does not implement any non-invasive debug functionality. For a description of the relationship to authentication interface signals, see the *CoreSight Architecture Specification v1.0*.

2.1.9 Test

Clock gating can be directly controllable during test mode. The following signals are provided, as Figure 2-3 shows:

- **DFTTESTMODE** enables the clock
- **DFTCLKDISABLE** disables the clock to reduce maximum power consumption during test.

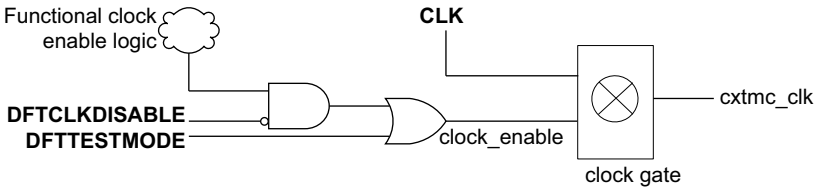


Figure 2-3 Test mode clock control

2.1.10 Synchronisation request interface

The TMC implements a synchronisation request slave interface, **SYNCREQS**. This is an output signal that is asserted for one cycle periodically when trace capture is enabled, and this period is controlled by the *Periodic Synchronization Counter Register* (PSCR). It is normally left unconnected, but can be used to control the synchronization frequency of trace sources.

In ETF configuration, the TMC also implements a synchronisation request master interface, **SYNCREQM**. This is an input signal, and **SYNCREQS** is driven by this signal instead of the Periodic Synchronization Counter if pulses on **SYNCREQM** occur more frequently. It is normally tied LOW.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.2 Operation

This section gives an overview of the TMC operation. The following sections describe the operation of architectural state machine:

- *TMC architectural state machine*
- *Standard usage models for the TMC* on page 2-10
- *Formatter and stop sequence* on page 2-12
- *Trigger, flush and stop events* on page 2-13
- *Scatter-gather* on page 2-16.

For more information, see Chapter 3 *Programmers Model*.

2.2.1 TMC architectural state machine

The TMC has an architectural state machine. Figure 2-4 shows the transactions of the state machine. The states are defined mainly by the Trace Capture Enable bit in the CTL Register, and the TMCReady bit in the STS Register.

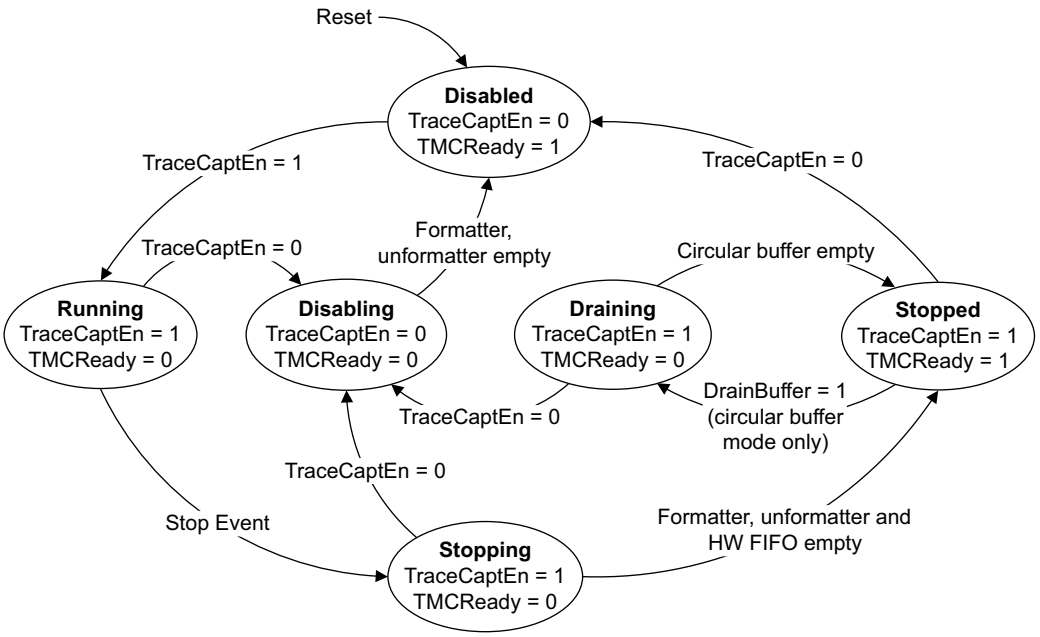


Figure 2-4 TMC architectural state machine

The following sections describe the states:

- *Disabled*
- *Running* on page 2-9
- *Stopping* on page 2-9
- *Stopped* on page 2-9
- *Draining* on page 2-9
- *Disabling* on page 2-10.

Disabled

This state is entered after reset, and, ultimately, whenever TraceCaptEn is cleared. You must perform all programming in this state.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

In this 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, provided that it was captured with scatter-gather disabled. You must manage the read pointer manually. This usage is deprecated in favor of reading the contents of the Circular Buffer while in the Stopped state instead.

The Running state is entered from this state by setting TraceCaptEn.

Running

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

The Stopping state is entered from this state when a Stop Event occurs. The Stop events are:

- a flush caused by any reason other than a request on **AFVALIDM** completes, and the StopOnFl bit in the FFCR is set
- a Trigger Event occurs, and the StopOnTrigEvt bit in the FFCR is set.

Stopping

In this state, the TMC begins to drain its contents. From the programmer's model, this state is indistinguishable from the Running state.

The Stopped state is entered from this state when:

- A Stop sequence has been written to the end of the trace. See *Standard usage models for the TMC* on page 2-10.
- The formatter and write buffer are empty.
- In Hardware FIFO mode, the FIFO and unformatter are empty.

Note

In Software FIFO mode, it might be necessary to read additional data from the FIFO to make space for the remaining data, such as the Stop sequence, to be written to it. If no such reads are performed, the Stopped state might never be reached.

Stopped

In this state, no more trace capture takes place, but data still in the TMC can be read out. When in this state:

- In Circular Buffer mode, the contents of the captured buffer can be read out over APB.
- In Circular Buffer mode, a drain can be initiated by setting the DrainBuffer bit in the FFCR. See *Draining*.
- In Software FIFO mode, the remaining contents of the FIFO can be read out over APB.

The Disabled state is entered from this state by clearing the TraceCaptEn bit.

Draining

In this state, the contents of the buffer captured in Circular Buffer mode are drained over the ATB interface. The TMC returns to the Stopped state as soon as the buffer is empty.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Disabling

This is an emergency stop state that can be entered at any time by clearing the TraceCaptEn bit. It differs from the Stopping state as follows:

- The TMC does not attempt to empty the contents of the FIFO in Hardware FIFO mode. Trace not yet output on the ATB master interface is lost.
- The next transition is to the Disabled state rather than the Stopped state. This means that:
  - in Circular Buffer mode, a drain operation is not possible, and the buffer contents are lost if they were captured with scatter-gather operation enabled
  - in Software FIFO mode, trace not yet retrieved is lost.
- Exit from this state is not dependent on reads being performed from the RRD Register, in Software FIFO mode, or writes on the ATB master interface being accepted in Hardware FIFO mode. If the FIFO is full, existing data is overwritten to enable the Stop sequence to complete.

In an ETR configuration, transition to the Disabled state can still be delayed indefinitely if the AXI master interface stalls indefinitely. In this case, it is necessary to clear the source of the stall because it is not possible to abort an active AXI transaction.

2.2.2 Standard usage models for the TMC

This section describes the recommended usage model for the TMC in the following operating modes:

- *Circular Buffer mode*
- *Software FIFO mode* on page 2-11
- *Hardware FIFO mode* on page 2-11
- *Detecting the end of trace data when formatting is disabled* on page 2-12.

Circular Buffer mode

The recommended standard usage model in this mode is as follows:

1. Wait until TMCReady is equal to one.
2. Program the MODE Register for Circular Buffer mode.
3. Program the FFCR Register. ARM recommends that you set the TrigOnTrigIn, FOnTrigEvt, StopOnFl, EnTI, and EnFt bits. This enables formatting, inserting a trigger when a trigger is observed on **TRIGIN**, and following a delay corresponding to the value of the TRG Register, flushing and then stopping the TMC.
4. Program the TRG Register, to control the amount of buffer to be dedicated to the period after a trigger is observed.
5. Set the TraceCaptEn bit in the CTL Register. This starts the trace session.
6. Wait until TMCReady is equal to one. This indicates that the trace session is over.
7. Read the contents of the trace buffer by performing successive reads to the RRD Register, until the value `0xFFFFFFFF` is returned.
8. Clear the TraceCaptEn bit in the CTL Register.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Software FIFO mode

The recommended standard usage model in this mode is as follows:

1. Wait until TMCReady is equal to one.
2. Program the MODE Register for Software FIFO mode.
3. Program the BUFWM Register to the required threshold fill level. You can usually set this register to zero.
4. Set the TraceCaptEn bit in the CTL Register. This starts the trace session.
5. Start reading data from the RRD Register. If the value `0xFFFFFFFF` is returned, then no data is available, and the read must be retried. Continue until the trace session is over, for example, following receipt of a trigger in the trace stream.
6. Set the StopOnFl bit in the FFCR Register.
7. Set the FlushMan bit in the FFCR Register. This flushes, then stops the TMC.
8. Read data from the FIFO to get flushed data, retrying when `0xFFFFFFFF` is returned, until TMCReady is equal to one. This indicates that all data has been written to the FIFO.

   ```
   Repeat {Read Data from RRD registerIf (Data = 0xFFFFFFFF) {Read TMCReady from STS
   registerIf (TMCReady = 1) {Stop}}   Else {  Add Data to the end of the trace}}
   ```

9. Read the remaining data from the FIFO, stopping when `0xFFFFFFFF` is returned. This indicates that the FIFO is empty.
10. Clear the TraceCaptEn bit in the CTL Register.

Hardware FIFO mode

The recommended standard usage model in this mode is as follows:

1. Wait until TMCReady is equal to one.
2. Program the MODE Register for Hardware FIFO mode.
3. Program the FFCR Register, setting the EnFT and EnTI bits. It is not usually necessary to set any other bits in this mode, because the trace sink performs flush and stop control.
4. Program the BUFWM Register to the required threshold fill level. You can usually set this register to zero.
5. Set the TraceCaptEn bit in the CTL Register.
6. The TMC is now operating in Hardware FIFO mode. Wait until the trace session is over, for example because the trace sink has stopped.
7. Set the StopOnFl bit in the FFCR Register.
8. Set the FlushMan bit in the FFCR Register. This flushes, then stops the TMC.
9. Wait until TMCReady is equal to one. The TMC is now empty.
10. Clear the TraceCaptEn bit in the CTL Register.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Detecting the end of trace data when formatting is disabled

The formatting protocol ensures that a continuous stream of 1s never occurs in a formatted trace stream. When formatting is enabled, that is EnFt bit in the FFCR register is set, and the trace memory is empty, the TMC returns `0xFFFFFFFF` for RRD reads to indirectly indicate that there is no valid trace data to read.

However, when formatting is disabled, that is, the EnFt bit in the FFCR Register is cleared, it is possible that a continuous stream of 1s appears in the non-formatted trace stream. In this case, you must combine the standard use cases that *Standard usage models for the TMC* on page 2-10 lists with the following techniques:

- The CBUFLEVEL Register indicates the current fill level of the trace buffer, and is valid when TraceCaptEn is equal to one, and when scatter-gather operation is disabled. You can use the value returned by this register to ensure that the RRD is read-only when trace memory has valid data.
- In scatter-gather operation, the CBUFLEVEL Register is not valid. However, the Empty bit in the STS Register is valid while TraceCaptEn is equal to one. You can use the Empty bit to ensure that the RRD is read-only when trace memory has valid data.

Note

This requires a read of the STS Register before every RRD read.

In practice, because of the low overhead of the formatter, 6%, it is expected that most usage models enable the formatter at all times.

2.2.3 Formatter and stop sequence

When EnFt in the FFCR is set, formatting is enabled. For more information about the formatting protocol, see the *CoreSight Architecture Specification*.

Depending on the configuration of the TMC, trace might be written up to 256 bits at a time. Additionally, when the formatter is enabled by setting the EnFt bit in the FFCR, a whole number of frames must be written. When stopping trace capture, the TMC pads the end of the trace so that every byte of trace that has been accepted by the TMC is written.

Formatter enabled

If the formatter is enabled when trace capture is stopped, then the traces are padded in the formatted frames with additional bytes of data with a value of `0x00` and an ID of `0x00`, until the following conditions are met:

- A whole number of frames have been generated.
- The trace is aligned to the memory width. This means that if the ATB interface is configured to 128 bits, then a multiple of two frames has been generated to meet the 256-bit memory width.

In Hardware FIFO mode, the ATB master interface does not generate any additional traces at the end of trace. However, all trace stored internally is flushed out of the TMC when trace capture is stopped.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Formatter disabled

Disabling the formatter is deprecated, and is supported in Circular Buffer mode only. If the formatter is disabled when trace capture is stopped, then the trace is padded with additional bytes so that the precise end of the trace can be determined, as follows:

- a single byte of value `0x01`, to indicate the position of the last byte before the stop sequence
- zero of more bytes of `0x00`, to align to the memory width.

Flexibility in the length of the stop sequence

The stop sequence might be longer than required to meet the above rules, to simplify the implementation.

2.2.4 Trigger, flush and stop events

The *Formatter and Flush Control Register* (FFCR), includes controls for the following:

- enabling the formatter to wrap data from multiple trace sources into frames *CoreSight Architecture Specification v1.0* describes
- the insertion of trigger markers into the formatted trace stream
- when to stop trace capture
- when to perform a flush
- draining the trace buffer when in Circular Buffer mode and in the Stopped state.

In Hardware FIFO mode and Software FIFO mode, setting the following bits of the FFCR results in Unpredictable behavior:

- TrigOnTrigEvt
- StopOnTrigEvt
- FOnTrigEvt.

TRIGIN and ATB slave interface trigger

Figure 2-5 on page 2-14 shows the actions taken when an event is sampled on **TRIGIN**, or a packet is accepted on the ATB slave interface when **ATID** is equal to `0x7D`.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

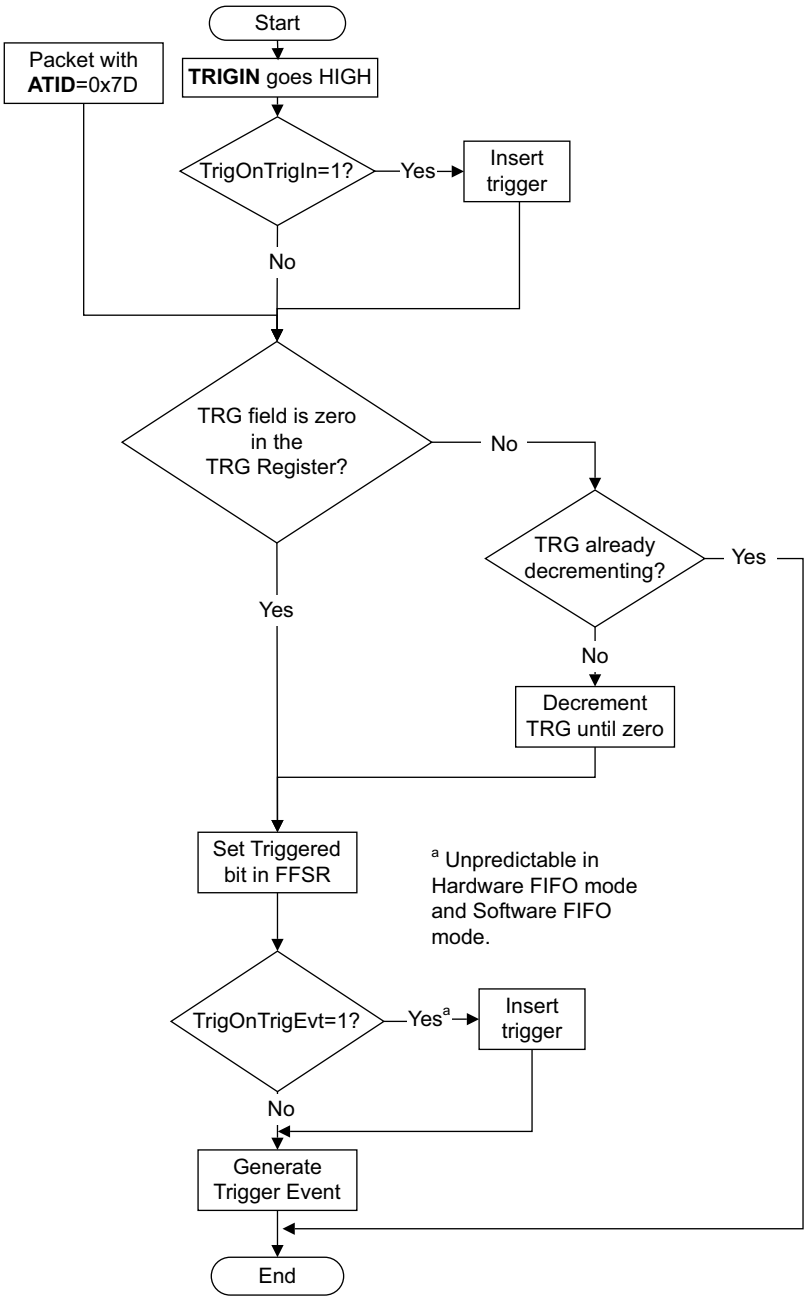


Figure 2-5 Actions taken on input trigger

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Flush and stop

The Trigger Event is one of several events that can cause various kinds of flush, stop the TMC, and cause additional trigger insertion. Table 2-2 shows how the TMC response to these events.

Table 2-2 Event generation

Event	MODE	Outcome
In ETF configuration, a flush request on the ATB master interface	CB or SWF	Acknowledged immediately, and otherwise is ignored.
	HWF	Flush the trace sources that feed the TMC, and the contents of the TMC.
FlushMan or (FOnFlIn & **FLUSHIN**)	CB	Flush the trace sources that feed the TMC. If TrigOnFl is set in the FFCR, insert a trigger. If StopOnFl is set in the FFCR, stop trace capture.
	HWF	Flush the trace sources that feed the TMC. If TrigOnFl is set in the FFCR, insert a trigger. If StopOnFl is set in the FFCR, stop trace capture.
	SWF	Flush the trace sources that feed the TMC, and ensure that the flushed trace is ready to be read by subsequent reads of the RRD register. If TrigOnFl is set in the FFCR, insert a trigger. If StopOnFl is set in the FFCR, stop trace capture.
Trigger Event	CB	If StopOnTrigEvt is set in the FFCR, then stop trace capture. If StopOnTrigEvt is not set in the FFCR: • Insert a trigger if TrigOnFl is set in the FFCR. • Stop trace capture if StopOnFl is set in the FFCR.
	HWF or SWF	Ignored.
TraceCaptEn bit cleared during trace capture	Any	Stop trace capture immediately. The captured trace is lost.

Note

The TMC always outputs any outstanding triggers in the trace before completing a flush or stopping trace capture. A rapid stream of triggers on **TRIGIN** or on the ATB slave interface can cause the flush or stop to be delayed.

Common usage

Many bits can be set simultaneously, leading to a wide range of programming settings, not all of which are useful. In practice, the most common setting in Circular Buffer mode is to set the TrigOnTrigIn, FOnTrigEvt, StopOnFl, EnTI and EnFt bits in the FFCR.

1. Wait for a trigger.
2. Insert a trigger into the trace stream.
3. Count down the trigger counter.
4. Flush.
5. Stop trace capture.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.2.5 Scatter-gather

When scatter-gather mode is enabled, the TMC uses tables in memory to find the addresses of pages of memory to use for trace storage.All pages are 4KB in size, and each is referenced by a single entry in the tables. The tables form a linked list, to ensure that the tables themselves not required to be larger than 4KB.

Figure 2-6 shows the format of the scatter-gather tables.

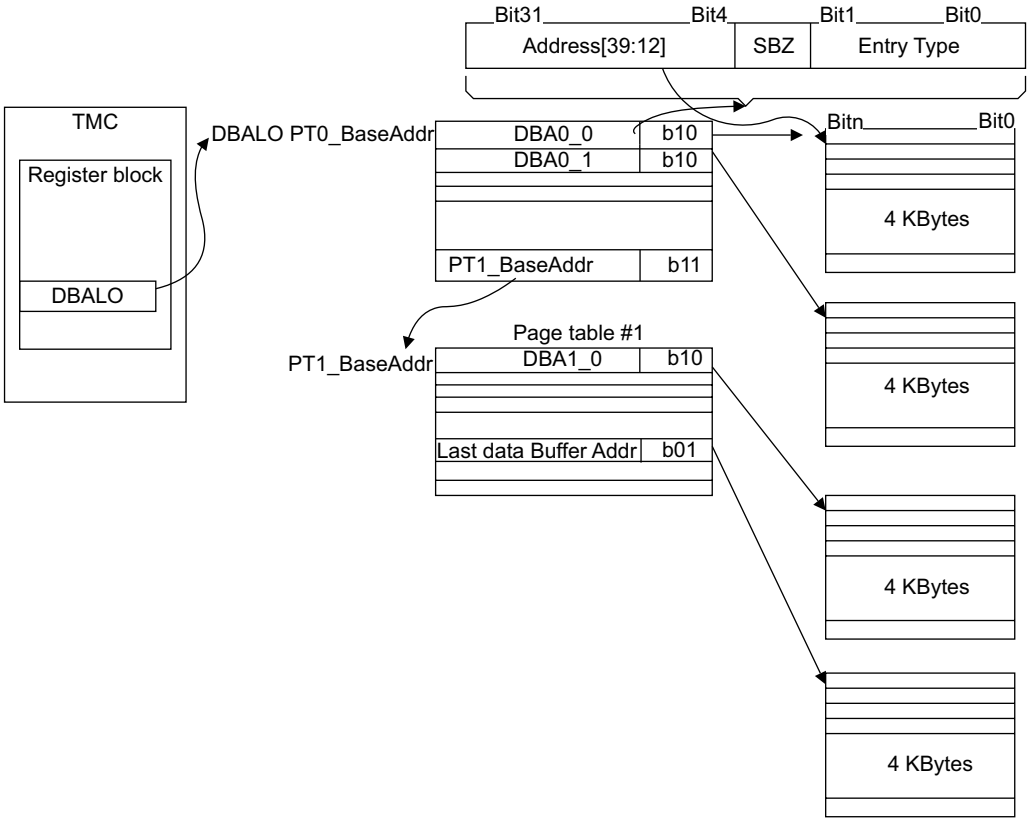


Figure 2-6 Scatter-gather description

Table 2-3 shows the format of the 32-bit entries used in this table.

Table 2-3 Page table 0 entries during scatter-gather operation

Bits	Description	
[31:4]	Bits [39:12] of an address. Bits [11:0] are always zero. The meaning of this field is dependent on the entry type.	
[3:2]	Reserved, SBZ	
[1:0]	Entry type:	
	00	Reserved
	01	Last. The address points to a 4KB page to use for buffer storage, and this is the last entry in the tables.
	10	Normal. The address points to a 4KB page to use for buffer storage, and the next entry in the table is valid. The entry must not be the last entry in a 4KB page.
	11	Link. The address points to the base of a further 4KB page of table entries.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

The TMC fetches each page table entry while the previous page is being written to, so that page table reads do not impact performance. When the last entry is reached, the TMC fetches the first entry from the first page table again.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.3 Drive a streaming interface using TMC

In ETR configuration, the TMC can drive trace over a streaming interface as *ETR to alternative interface* on page 1-7 describes. The following sections describe:

- *Programming*
- *Integration*
- *Unsupported modes of operation* on page 2-20.

2.3.1 Programming

When the TMC is driving a streaming device, the registers must be programmed as follows:

RSZ

The RSZ Register must be programmed with a value equal to the width of the ATB interface. For example, in a 32-bit ATB configuration, the AXI master interface is also 32 bits wide, and the RSZ Register must contain the value `0x00000001`, corresponding to a four byte buffer. This causes all writes to be made to the same address.

MODE

The MODE Register must be set to Circular Buffer mode.

AXICTL

The AXICTL Register must be set as follows:

- WrBurstLen is equal to one
- scatter-gather operation disabled
- other bits as required by the streaming device.

DBALO or DBAHI

These must be set to hold the AXI address of the streaming device data register.

FFCR

Since the streaming device cannot embed trigger data directly, it is recommended that the EnFt and EnTI bits of the FFCR are set, even if only one trace source is enabled. The setting of the other bits depends on the usage model.

2.3.2 Integration

Figure 2-7 on page 2-19 shows how the TMC can be integrated with a streaming device conforming to an architecture such as HSSTP.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

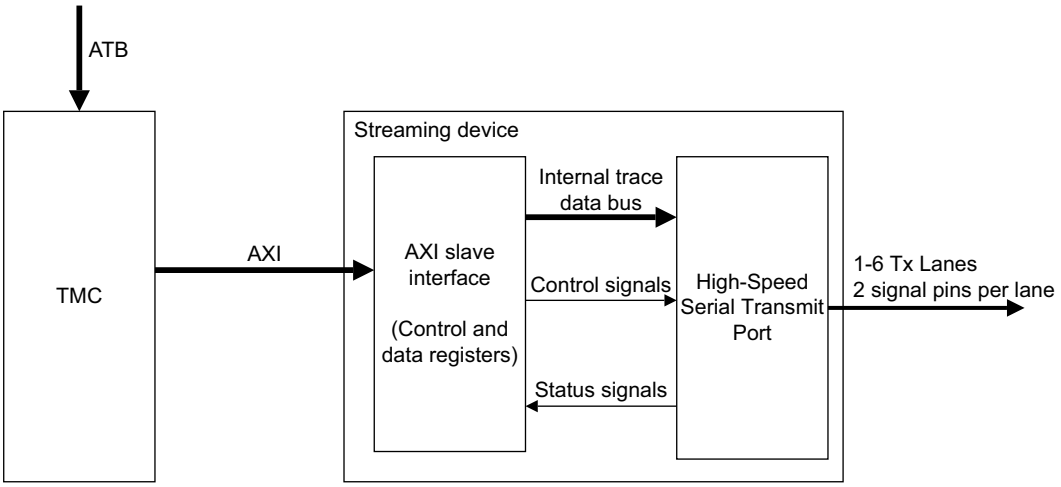


Figure 2-7 Integrating the TMC with HSSTP

The streaming device can use its AXI slave interface to manage the flow of trace. A stall on the AXI slave interface propagates back through the trace system, and ultimately to the trace sources, such as an ETM or STM.

Frame alignment considerations

The streaming device must output trace data in frame multiples, that is, multiples of 16 bytes. This ensures that the device receiving the trace from the streaming device can tell where each frame begins.

Treatment of bursts

When the TMC is programmed as *Integration* on page 2-18 describes, each AXI burst is guaranteed to start on a frame boundary. It is recommended that each packet in the protocol implemented by the streaming device corresponds to one AXI burst. This ensures:

- That the device connected to the transmit port can identify the alignment of frames. This is the only way to determine frame alignment, because the TMC does not include separate synchronization packets to indicate frame alignment.
- That the streaming device does not hold trace at the end of the trace run. When trace capture is complete, all trace is flushed from the TMC. It would be possible for the streaming device to temporarily store several AXI bursts to build a larger, more efficient packet for transmission. However, this would risk some trace remaining in the streaming device when trace capture is disabled. By outputting one packet per burst, no trace is stored in the streaming device for a significant period of time.

Direct or shared connection

You can connect the TMC to the streaming device either directly as *Integration* on page 2-18 describes, or over a shared interconnect. For example, if the streaming device contains programming registers that must be programmed through its AXI slave interface, then this interface must be accessible to the debugger and on-chip software, for example, by bridging from the debug APB bus. By connecting the TMC to a shared AXI bus, the same instance can be used to drive a streaming interface, or to store data in on-chip memory.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.3.3 Unsupported modes of operation

The TMC only supports streaming to an AXI slave. It does not support controllers that are a bus master, and work by reading blocks of data from memory on request. For low-speed interfaces, it is possible to implement a controller that periodically attempts to read data from an ETB configuration of the TMC, programmed in Software FIFO mode, using the APB interface. However, the TMC has not been optimized for this usage model, and it is likely to be slow, that is, up to seven APB cycles per read.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.4 Backwards compatibility

The ETB configuration of the CoreSight TMC is designed to support many usage models, including those supported by CoreSight ETB, with changes to the programmer's model minimized wherever possible. This section is provided to support tools providers with upgrading support from the CoreSight ETB to the TMC in ETB configuration.

This section describes the differences in programmer's model between the CoreSight ETB and the TMC when:

- registers new to the TMC have their reset values
- the TMC is configured in ETB configuration with a 32-bit ATB interface.

This section does not describe the effects of new functionality available only in the TMC. Instead, it describes the changes that are likely to be required to existing debug tools with support for the CoreSight ETB so that they can support the same features in the TMC, in Circular Buffer mode with scatter-gather operation disabled.

The following considerations are described:

- *ID registers*
- *RRP or RWP base*
- *Restrictions while in stopped state*
- *Starting and stopping trace capture* on page 2-22
- *AcqComp, FtEmpty, and FtStopped* on page 2-22
- *Wider memory interface* on page 2-22
- *RWD Register is WO* on page 2-22
- *Empty bit* on page 2-22
- *TRG Register* on page 2-23.

2.4.1 ID registers

The ID registers of the TMC have been updated to reflect the fact that it is a new design. Existing debug tools are required to modify to recognize the new part number, `0x961`, and must read the DEVID Register to determine whether the TMC configuration is the one that they support.

2.4.2 RRP or RWP base

In the CoreSight ETB, the RRP and RWP are expressed in 32-bit words. For example, address `0x00000000` points to the first word of memory, and address `0x00000001` points to the second word of memory, four bytes later.

In the TMC, the RRP and RWP are expressed as byte addresses. For example, the first and second 32-bit words are at address `0x00000000` and `0x00000004` respectively.

The reason for this change is to provide greater consistency with the ETR configuration.

Existing debug tools are required to modify to take account of this change. It is recommended that tools move to the simplified sequence of operations that *Circular Buffer mode* on page 2-10 describes that do not use the RRP and RWP registers.

2.4.3 Restrictions while in stopped state

Writing to the following registers causes unpredictable behavior in Circular Buffer mode in the Stopped state in the TMC, but are permitted in the Stopped state in the CoreSight ETB.

- RRP
- RWP

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

- RWD
- TRG
- the EnFt and EnTI bits of the FFCR.

Debug tools must instead transition to the Disabled state before writing to these registers, as *Circular Buffer mode* on page 2-10 describes.

Note

It is not necessary to write to the RRP and RWP registers in most usage models because the pointers are managed automatically. When EnFt is set, debug tools can read the contents of the Circular Buffer in Stopped state solely by successive reads of the RRD Register.

2.4.4 Starting and stopping trace capture

In the CoreSight ETB, the StopOnTrigEvt or StopOnFl bits of the FFCR behave largely independently to the TraceCaptEn bit, and must sometimes be cleared and set again to enable trace capture to restart.

In the TMC, a strict state machine is followed as *TMC architectural state machine* on page 2-8 describes. The StopOnTrigEvt and StopOnFl bits are used purely for programming and it is not necessary to toggle them as part of a trace capture restart sequence.

2.4.5 AcqComp, FtEmpty, and FtStopped

In the CoreSight ETB, these signals behaved differently, but were all used in a similar way, to identify when trace capture was complete. In the CoreSight TMC, these signals have similar behavior. The AcqComp bit has been renamed TMCReady, and when trace capture is in progress, is the bit that you use to determine when trace capture is complete. The FtEmpty and FtStopped register bits are retained for backwards-compatibility, but their use is deprecated.

2.4.6 Wider memory interface

Accesses to the RRD and RWD registers must be made sequentially according to the width of the memory interface. For example, a 32-bit ATB interface has a 64-bit memory interface, and therefore RRD and RWD accesses must be made in pairs. The RRP and RWP are only to be updated when a full set of accesses has been performed.

This is not likely to affect existing debug tools because:

- the RWD register is for integration testing purposes only, and is not required in normal use
- the common usage model for the RRD Register is to read it multiple times in succession to download the Circular Buffer contents, and this usage model is unaffected.

2.4.7 RWD Register is WO

In the CoreSight ETB, the RWD Register was read-write. In the TMC it is write-only.

2.4.8 Empty bit

The Empty bit has been added to the STS Register. Existing debug tools can ignore the value of this bit.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

2.4.9 TRG Register

The TRG Register is reset to zero when TMCReady transitions from clear to set, that is, on entry to the Stopped or Disabled states. If the trigger count has started but not reached zero at this point, then no trigger events are observed.

The CoreSight ETB did not automatically clear the TRG Register at any time.

Tools must still reload the TRG Register between each trace capture session. Standard usage models are not affected by this change.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Chapter 3
Programmers Model

This chapter describes the programmers model. It contains the following sections:

This chapter is auto-generated from IP-XACT code.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.1 About this programmers model

The following information applies to the registers:

- The base address 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 a logic 0 by a system or power-on reset.
- Access type in Table 3-1 on page 3-3 is described as follows:

  RW Read and write.

  RO Read only.

  WO Write only.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.2 Register summary

Table 3-1 shows the registers in base offset order.

Table 3-1 TMC register summary

Offset	Name	Type	Reset	Description
0x004	RSZ	RO, RW[a]	0x00000080[b]	*RAM Size Register* on page 3-5
0x00C	STS	RO	0x0000000C	*Status Register* on page 3-6
0x010	RRD	RO	-	*RAM Read Data Register* on page 3-8
0x014	RRP	RW	0x00000000	*RAM Read Pointer Register* on page 3-9
0x018	RWP	RW	0x00000000	*RAM Write Pointer Register* on page 3-11
0x01C	TRG	RW	0x00000000	*Trigger Counter Register* on page 3-12
0x020	CTL	RW	0x00000000	*Control Register* on page 3-13
0x024	RWD	WO	-	*RAM Write Data Register* on page 3-13
0x028	MODE	RW	0x00000000	*Mode Register* on page 3-14
0x02C	LBUFLEVEL	RO	0x00000000	*Latched Buffer Fill Level* on page 3-15
0x030	CBUFLEVEL	RO	0x00000000	*Current Buffer Fill Level* on page 3-16
0x034	BUFWM	RW	0x00000000	*Buffer Level Water Mark* on page 3-16
0x038	RRPHI	RW	0x00000000	*RAM Read Pointer High Register* on page 3-17
0x03C	RWPHI	RW	0x00000000	*RAM Write Pointer High Register* on page 3-18
0x110	AXICTL	RW	0x00000000	*AXI Control Register* on page 3-18
0x118	DBALO	RW	0x00000000	*Data Buffer Address Low Register* on page 3-20
0x11C	DBAHI	RW	0x00000000	*Data Buffer Address High Register* on page 3-21
0x300	FFSR	RO	0x00000002	*Formatter and Flush Status Register* on page 3-21
0x304	FFCR	RW	0x00000000	*Formatter and Flush Control Register* on page 3-22
0x308	PSCR	RW	0x00000000	*Periodic Synchronization Counter Register* on page 3-25
0xED0	ITATBMDATA0	WO	-	*Integration Test ATB Master Data Register 0* on page 3-26
0xED4	ITATBMCTR2	RO	-	*Integration Test ATB Master Interface Control 2 Register* on page 3-28
0xED8	ITATBMCTR1	WO	-	*Integration Test ATB Master Control Register 1* on page 3-29
0xEDC	ITATBMCTR0	WO	-	*Integration Test ATB Master Interface Control 0 Register* on page 3-29
0xEE0	ITMISCOP0	WO	-	*Integration Test Miscellaneous Output Register 0* on page 3-30
0xEE8	ITTRFLIN	RO	-	*Integration Test Trigger In and Flush In Register* on page 3-31
0xEEC	ITATBDATA0	RO	-	*Integration Test ATB Data Register 0* on page 3-32
0xEF0	ITATBCTR2	WO	-	*Integration Test ATB Control 2 Register* on page 3-34
0xEF4	ITATBCTR1	RO	-	*Integration Test ATB Control 1 Register* on page 3-35
0xEF8	ITATBCTR0	RO	-	*Integration Test ATB Control 0 Register* on page 3-35

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-1 TMC register summary (continued)

Offset	Name	Type	Reset	Description
0xF00	ITCTRL	RW	0x00000000	*Integration Mode Control Register* on page 3-36
0xFA0	CLAIMSET	RW	0x0000000F	*Claim Tag Set Register* on page 3-37
0xFA4	CLAIMCLR	RW	0x00000000	*Claim Tag Clear Register* on page 3-38
0xFB0	LAR	WO	-	*Lock Access Register* on page 3-38
0xFB4	LSR	RO	0x00000003	*Lock Status Register* on page 3-39
0xFB8	AUTHSTATUS	RO	-	*Authentication Status Register* on page 3-40
0xFC8	DEVID	RO	-	*Device Configuration Register* on page 3-40
0xFCC	DEVTYPE	RO	-	*Device Type Identifier Register* on page 3-42
0xFD0	PERIPHID4	RO	0x00000004	*Peripheral ID4 Register* on page 3-42
0xFD4	PERIPHID5	RO	0x00000000	*Peripheral ID5 Register* on page 3-43
0xFD8	PERIPHID6	RO	0x00000000	*Peripheral ID6 Register* on page 3-43
0xFDC	PERIPHID7	RO	0x00000000	*Peripheral ID7 Register* on page 3-44
0xFE0	PERIPHID0	RO	0x00000061	*Peripheral ID0 Register* on page 3-44
0xFE4	PERIPHID1	RO	0x000000B9	*Peripheral ID1 Register* on page 3-45
0xFE8	PERIPHID2	RO	0x0000001B	*Peripheral ID2 Register* on page 3-45
0xFEC	PERIPHID3	RO	0x00000000	*Peripheral ID3 Register* on page 3-46
0xFF0	COMPID0	RO	0x0000000D	*Component ID0 Register* on page 3-47
0xFF4	COMPID1	RO	0x00000090	*Component ID1 Register* on page 3-47
0xFF8	COMPID2	RO	0x00000005	*Component ID2 Register* on page 3-48
0xFFC	COMPID3	RO	0x000000B1	*Component ID3 Register* on page 3-48

a. RO for ETB and ETF configurations and RW for ETR configuration.
b. Reset to zero in ETR configuration. Reflects the RAM size in ETB and ETF configurations.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3 Register descriptions

This section describes the TMC registers.

3.3.1 RAM Size Register

The RSZ Register characteristics are:

Purpose Defines the size, in 32-bit words, of the local RAM buffer.

ETB, ETF configurations

This value is configurable in the RTL MEM_SIZE parameter, but fixed at synthesis. Supported size is in powers of two only.

ETR configuration

This register is programmable, and selects the size of the memory region to be used for trace storage. If the TMC is programmed for scatter-gather operation, the contents of the RSZ Register are ignored.

The size of the trace buffer must be a multiple of the AXI data width. This can be found by looking at the MEMWIDTH field in the DEVID Register.

The maximum trace buffer size permitted is 4GB. The minimum trace buffer size enabled in Software FIFO mode and Hardware FIFO mode is 512 bytes. The minimum trace buffer size enabled in Circular Buffer mode is one AXI dataword. This is useful when writing to a streaming interface. See *Drive a streaming interface using TMC* on page 2-18.

The burst length programmed in the AXICTL Register, WrBurstLen, must be compatible with the trace buffer size and the AXI data width so that the total number of bytes of data transferred in a burst is not greater than the trace buffer size. Programming an incompatible burst length results in Unpredictable behavior.

Modifying this register when the TMCReady bit, STS Register, `0x00C` is clear, or the TraceCaptEn bit, CTL Register, `0x020` is set, results in Unpredictable behavior.

Configurations Present in all configuration.

Figure 3-1 shows the RSZ Register bit assignments.

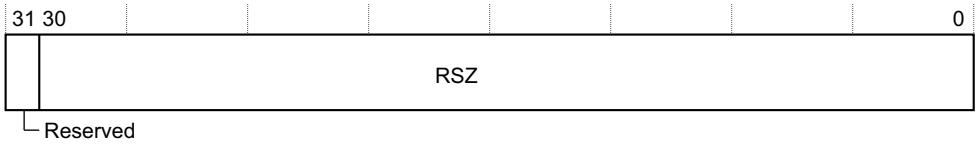


Figure 3-1 RSZ Register bit assignments

Table 3-2 shows the RSZ Register bit assignments.

Table 3-2 RSZ Register bit assignments

Bits	Name	Function
[31]	Reserved	Reserved.
[30:0]	RSZ	Size of the RAM in 32-bit words. For example, for 1KB RAM, this register is `0x00000100`. For 4GB RAM, this register is `0x40000000`.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.2 Status Register

The STS Register characteristics are:

Purpose TMC Status register.

Configurations Present in all configuration.

Figure 3-2 shows the STS Register bit assignments.

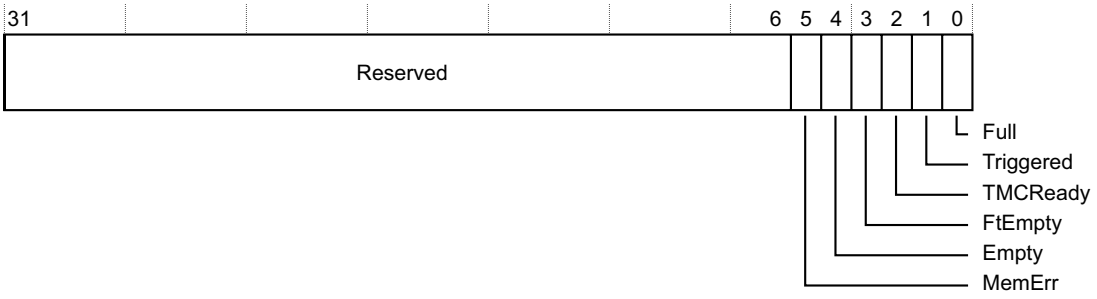


Figure 3-2 STS Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-3 shows the STS Register bit assignments.

Table 3-3 STS Register bit assignments

Bits	Name	Function
[31:6]	Reserved	Reserved.
[5]	MemErr	This bit indicates that an error has occurred on the AXI master interface. The error could be because of an error response from an AXI slave or because of the status of the Authentication interface inputs: • This bit is set whenever an AXI read or write response of type SLVERR (`b10`) or DECERR (`b11`) is encountered. These error types are not distinguished between, and EXOKAY (`b01`) is treated as OKAY (`b00`). An exclusive okay response is illegal because the TMC does not request any exclusive accesses. • This bit is also set instead of initiating an AXI access when Non-secure Invasive Debug is disabled, and instead of initiating a secure access when Secure Invasive Debug is disabled. For more information, see *Authentication Status Register* on page 3-40 and *AXI Control Register* on page 3-18. When the Memory Error Status bit is set: • No more AXI accesses are performed other than any commands in the internal pipeline. • Outstanding AXI read and write responses are ignored. Pending responses are counted, but the responses received are discarded. • A read access to the RRD Register or a write access to the RWD Register returns a slave error. This includes accesses to these registers that would otherwise have been ignored without returning a slave error, for example, when the TMC is enabled and in Circular Buffer mode or in Hardware Read FIFO mode. • The formatter stops, and sets TMCReady when it has stopped. • Reading the RWP, RWPHI, RRP, and RRPHI registers returns Unknown values. This bit is cleared by: • writing to this register with this bit set when TMCReady is HIGH • setting TraceCaptEn when TraceCaptEn is LOW.
[4]	Empty	If set, this bit indicates that the TMC does not contain any valid trace data in the trace memory. This does not, however, mean that the pipeline stages within the TMC are empty. To determine whether the pipeline stages within the TMC are empty, read the TMCReady bit. This bit is set on reset. This bit is valid only when TraceCaptEn is HIGH. This bit reads as zero when TraceCaptEn is LOW. **Note** In Circular Buffer mode, it is possible that the Empty bit and the Full bit in this register are one at the same time because the Full bit in this mode, when set, does not clear until TraceCaptEn is set.
[3]	FtEmpty	This bit is set when trace capture has stopped, and all internal pipelines and buffers have drained. Unlike TMCReady, it is not affected by buffer drains and AXI accesses.The **ACQCOMP** output reflects the value of this bit unless the TMC is in integration mode.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-3 STS Register bit assignments (continued)

Bits	Name	Function
[2]	TMCReady	This bit is set when all the following are true: • Trace capture has stopped and all internal pipelines and buffers have drained. This is equivalent to being in the Stopped or Disabled state. See *TMC architectural state machine* on page 2-8. • The TMC is not training because of the DrainBuffer bit of the FFCR being set. • In ETR configuration, the AXI interface is not busy. This case can be used to detect page table reads in scatter-gather mode when in Stopped state.
[1]	Triggered	The Triggered bit is set when trace capture is in progress and the TMC has detected a Trigger Event. For more information, see *Trigger, flush and stop events* on page 2-13. This bit is cleared when leaving Disabled state. This bit is operational only in the Circular Buffer mode. In all other modes, this bit is always LOW. This bit does not indicate that a trigger has been embedded in the formatted output trace data from the TMC. Trigger indication on the output trace stream is determined by the programming of the Formatter and Flush Control Register, FFCR, `0x304`.
[0]	Full	This bit can help to determine how much of the trace buffer contains valid data. **Circular Buffer mode** This flag is set when the RAM write pointer wraps around the top of the buffer, and remains set until the TraceCaptEn bit is cleared and set. **Software FIFO mode and Hardware FIFO mode** This flag indicates that the current space in the trace memory is less than or equal to the value programmed in the BUFWM Register, that is, Fill level >= MEM_SIZE - BUFWM. In the ETR configuration, if the TMC is programmed for scatter-gather operation, this bit indicates whether the Trace memory is currently full regardless of the value programmed in the BUFWM Register. **Note** In scatter-gather mode, the TMC might be unable to accept new trace without this bit being set, if a read from the scatter-gather tables is delayed sufficiently. When entering Disabled state, this bit retains its previous value. When exiting Disabled state, this bit is cleared. The **FULL** output from the TMC reflects the value of this register bit, except when the Integration Mode bit in the ITCTRL Register, `0xF00`, is set.

3.3.3 RAM Read Data Register

The RRD Register characteristics are:

Purpose Reading this register enables data to be read from the trace memory. When the memory width given in the DEVID register is greater than 32 bits, multiple reads to this register must be performed together to read a full memory width of data. For example, if the memory width is 128 bits, then reads from this register must be performed four at a time. When a full memory width of data has been read, the RAM Read Pointer is incremented to the next memory word.When no data is available, this

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

register returns 0xFFFFFFFF. This value is chosen because it cannot be generated as part of the trace data when the formatter is enabled. See *Standard usage models for the TMC* on page 2-10.

Trace Capture disabled

When in Disabled state, the TMC mode is ignored. Reading this register returns the contents of the Local RAM buffer or AXI memory at the location addressed by the RAM Read Pointer Register.

Circular Buffer mode

When in Stopped state in Circular Buffer mode and the buffer is not empty, reading this register returns the next word of data from the trace buffer. When all of the trace buffer has been read, the Empty bit in the STS Register is set, and more reads return 0xFFFFFFF.Reading this register when not in Stopped state returns 0xFFFFFFFF.

Software FIFO mode

Reading this register returns data from the FIFO. If this register is read when the FIFO is empty, the data returned is 0xFFFFFFFF.

Hardware FIFO mode

Reading this register returns 0xFFFFFFFF.

Reading this register alters the internal state of the TMC, and can only be performed if the device is unlocked. Reading this register when the device is locked returns 0xFFFFFFFF.

In the ETR configuration, when the MemErr bit in the STS Register is set, reading this register returns an error response on the APB slave interface.

Configurations Present in all configuration.

Figure 3-3 shows the RRD Register bit assignments.

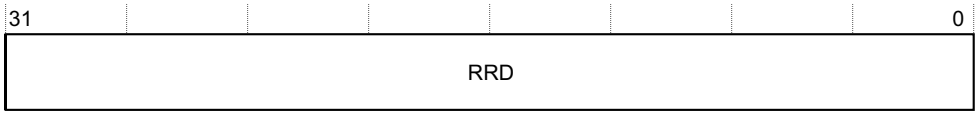


Figure 3-3 RRD Register bit assignments

Table 3-4 shows the RRD Register bit assignments.

Table 3-4 RRD Register bit assignments

Bits	Name	Function
[31:0]	RRD	Reads return data from Trace RAM

3.3.4 RAM Read Pointer Register

The RRP Register characteristics are:

Purpose The RAM Read Pointer Register contains the value of the read pointer that is used to read entries from the trace memory over the APB interface.

The value written to this register must be a byte-address aligned to the width of the trace memory databus and to a frame boundary. For example, for 64-bit wide trace memory and 128-bit wide trace memory, the four

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

LSBs must be 0s. For 256-bit wide trace memory, the five LSBs must be 0s. The width of the trace memory can be obtained by reading the MEMWIDTH field in the DEVID Register, `0xFC8`.

When one complete buffer or FIFO entry has been read through the RRD Register, the RAM Read Pointer Register is incremented by the number of bytes per memory width of data. For example, for 64-bit wide memory, it is incremented by eight. For 128-bit wide memory, it is incremented by 16 for every complete memory entry read. When the RAM Read Pointer is incremented after having reached its maximum value, it wraps around.

The width of this register in the ETB or ETF configurations is $\log_2$(MEM_SIZE*4). In the ETR configuration, this register is 32 bits wide, and the contents of this register represents the lower 32 bits of the 40-bit AXI address used to access trace memory. If scatter-gather operation is enabled, this register represents the next address in trace memory to be read, not the address of a page table entry.

When in Disabled state, a write to this register sets the value of the trace memory address from which data is fetched on a subsequent RRD read. A write to this register when not in Disabled state results in Unpredictable behavior.This register can be read:

- when in Disabled state
- when in Stopped state, in Circular Buffer mode
- when in Running, Stopping or Stopped states, in Software FIFO mode.

When entering Disabled state in Circular Buffer mode with scatter-gather mode disabled, this register points to the next location in the trace buffer to be read. This is for backwards compatibility purposes, so that the buffer can be read while in Disabled state. It is recommended that you read the buffer contents while in Stopped state instead, because the pointers are managed automatically.

Configurations Present in all configurations.

Figure 3-4 shows the RRP Register bit assignments.

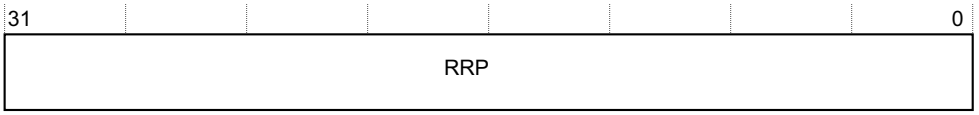


Figure 3-4 RRP Register bit assignments

Table 3-5 shows the RRP Register bit assignments.

Table 3-5 RRP Register bit assignments

Bits	Name	Function
[31:0][a]	RRP	This value represents the location in trace memory that is accessed on a subsequent RRD read

a. The width of this register is dependent on the TMC configuration.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.5 RAM Write Pointer Register

The RWP Register characteristics are:

Purpose The RWP Register sets the write pointer used to write entries into trace RAM.

Writing to this register other than when in Disabled state results in Unpredictable behavior. When in Disabled state, a write to this register sets the value of the trace memory address to which data is written on a subsequent RWD write.

The value written to this register must be a byte-address aligned to the width of the trace memory databus and to a frame boundary. For example, for 64-bit wide trace memory and 128-bit wide trace memory, the four LSBs must be 0s. For 256-bit wide trace memory, the five LSBs must be 0s. The width of the trace memory can be obtained by reading the MEMWIDTH field in the DEVID Register, `0xFC8`.

Reading this register returns the current memory location being referenced, to which the next write would occur. When one complete buffer or FIFO entry has been written to the RWD Register, the RAM Write Pointer Register is incremented by the number of bytes per memory width of data. For example, for 64-bit wide memory, it is incremented by eight. For 128-bit wide memory it is incremented by 16 for every complete memory entry write.

When this register wraps around its maximum value, the Full flag in the Status Register, STS, `0x00C`, is set.

In the ETB or ETF configurations, the width of this register is $\log_2$(MEM_SIZE*4). In the ETR configuration, this register is 32 bits wide, and the contents of this register represent the lower 32 bits of the 40-bit AXI address used to access trace memory.

In ETB and ETF configurations, when in Circular Buffer mode, this register can be used to set the address to start capturing data from. This is for backwards compatibility purposes, to enable the **FULL** signal to be generated before the buffer becomes full. In other configurations and modes, the RAM Write Pointer is reset to the start of trace memory when exiting Disabled state.This register can be read:

- when in Disabled state
- when in Stopped state, in Circular Buffer mode
- when in Running, Stopping or Stopped states, in Software FIFO mode.

Configurations Present in all configurations.

Figure 3-5 shows the RWP Register bit assignments.

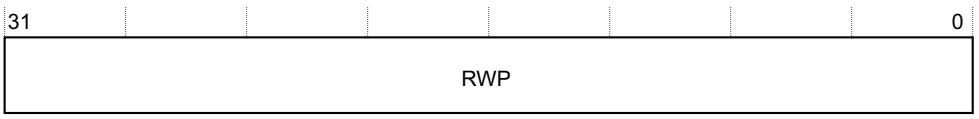


Figure 3-5 RWP Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-6 shows the RWP Register bit assignments.

Table 3-6 RWP Register bit assignments

Bits	Name	Function
[31:0][a]	RWP	This value represents the location in trace memory that are accessed on a subsequent write to the trace memory

a. The width of this register is dependent on the TMC configuration.

3.3.6 Trigger Counter Register

The TRG Register characteristics are:

Purpose In Circular Buffer mode, specifies the number of 32-bit words to capture in the Trace RAM following the detection of either a rising edge on the **TRIGIN** input or a trigger packet in the incoming trace stream, **ATID** = `7'h7D`. On capturing the specified number of datawords, a Trigger Event occurs. The effect of a Trigger Event on TMC behavior is controlled by the FFCR Register, `0x304`. See *Trigger, flush and stop events* on page 2-13.

The number of 32-bit words written into the Trace RAM following the trigger is the value stored in this register plus one. This register is ignored when the TMC is in Software FIFO mode or Hardware FIFO mode.

When the trigger counter starts counting, any additional triggers, either on **TRIGIN** or in the incoming trace stream, are ignored until the counter reaches zero. When the trigger counter has reached zero, it remains at zero until it is re-programmed with a write to this register.

This register is cleared when TMCReady goes HIGH, so that the state of the counter when trace capture has stopped does not affect a subsequent trace capture session.

Writing to this register when not in Disabled state results in Unpredictable behavior.

A read access to this register is permitted at any time when in Disabled state, or in Circular Buffer mode. A read access returns the current value of the Trigger counter.

The width of this register and the Trigger counter depends on the size of the trace memory. In ETB and ETF configurations, the width of the counter is $\log_2$(MEMSIZE). The width of this register in ETR configuration is 32 bits.

Configurations Present in all configurations.

Figure 3-6 shows the TRG Register bit assignments.

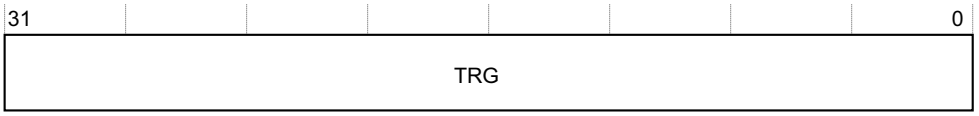


Figure 3-6 TRG Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-7 shows the TRG Register bit assignments.

Table 3-7 TRG Register bit assignments

Bits	Name	Function
[31:0][a]	TRG	This count represents the number of 32-bit words between a **TRIGIN** or trigger packet and a Trigger Event.

a. The width of this register is dependent on the TMC configuration.

3.3.7 Control Register

The CTL Register characteristics are:

Purpose Controls trace stream capture.

Configurations Present in all configurations.

Figure 3-7 shows the CTL Register bit assignments.

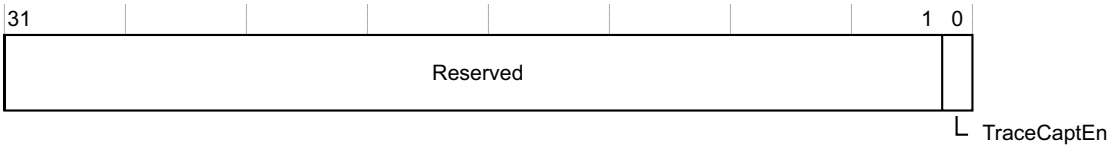


Figure 3-7 CTL Register bit assignments

Table 3-8 shows the CTL Register bit assignments.

Table 3-8 CTL Register bit assignments

Bits	Name	Function
[31:1]	Reserved	Reserved.
[0]	TraceCaptEn	Controls trace capture. See *TMC architectural state machine* on page 2-8.

3.3.8 RAM Write Data Register

The RWD Register characteristics are:

Purpose Enables testing of Trace RAM connectivity to the TMC.

When in Disabled state, a write to this register stores data at the location pointed to by the RWP. Writes to this register when not in Disabled state are ignored.When the memory width given in the DEVID register is greater than 32-bit, multiple writes to this register must be performed together to read a full memory width of data. For example, if the memory width is 128 bits, then writes to this register must be performed four at a time. When a full memory width of data has been written, the data is written to memory and the RAM Write Pointer is incremented to the next memory word.

In ETR configuration, when the MemErr bit in the STS Register is set, writing to this register returns an error response on the APB slave interface and the write data is discarded.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Writing to this register other than when in Disabled state and in integration mode results in Unpredictable behavior.

Configurations Present in all configurations.

Figure 3-8 shows the RWD Register bit assignments.

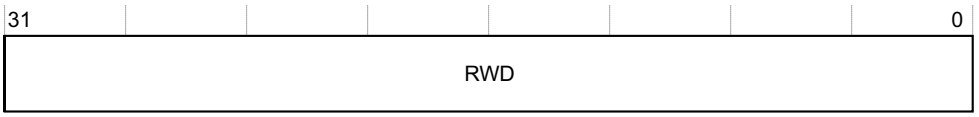


Figure 3-8 RWD Register bit assignments

Table 3-9 shows the RWD Register bit assignments.

Table 3-9 RWD Register bit assignments

Bits	Name	Function
[31:0]	RWD	Data written to this register is placed in the Trace RAM.

3.3.9 Mode Register

The MODE Register characteristics are:

Purpose Controls TMC operating mode.

When configured as an ETB or ETR, the TMC can operate in the following modes:

- Software FIFO mode
- Circular Buffer mode.

When configured as an ETF, the TMC has an additional mode of operation, Hardware FIFO mode.

The operating mode can be changed only when the TMC is disabled. Attempting to write to this register while not in Disabled state results in Unpredictable behavior. The operating mode is ignored when in Disabled state.

Configurations Present in all configurations.

Figure 3-9 shows the MODE Register bit assignments.

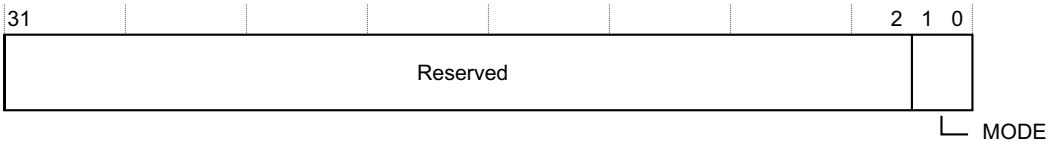


Figure 3-9 MODE Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-10 shows the MODE Register bit assignments.

Table 3-10 MODE Register bit assignments

Bits	Name	Function
[31:2]	Reserved	Reserved.
[1:0]	MODE	Selects the operating mode. The following are the operation modes: `3` **=Reserved** Reserved `2` **= Hardware FIFO mode** This mode is available only in the ETF configuration. Enabling this mode in any other configuration results in Unpredictable behavior. In this mode, the trace memory is used as a FIFO that is drained through the ATB master interface. Trace data is captured into the Trace RAM and when full, the incoming trace stream is stalled. When the Trace buffer is non-empty, trace data is drained out through the ATB master interface. `1` **= Software FIFO mode** In this mode, the trace memory is used as a FIFO that can be read through the RRD Register while trace is being captured. Trace data is captured into the Trace RAM and when full, the incoming trace stream is stalled. `0` **= Circular Buffer mode** In this mode, the trace memory is used as a Circular Buffer. Trace data is captured into the Trace memory starting from the location pointed to by the write pointer register. Even when the trace memory is full, incoming trace data continues to be overwritten into the trace memory until a stop condition has occurred. See *Formatter and Flush Control Register* on page 3-22 and *Control Register* on page 3-13. For more information about standard usage models in different modes, see *Standard usage models for the TMC* on page 2-10.

3.3.10 Latched Buffer Fill Level

The LBUFLEVEL Register characteristics are:

Purpose 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.

This register does not apply when the TMC is programmed for scatter-gather operation in ETR configuration. In this case, reading this register returns `0x00000000`.

When entering Disabled state, this register 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 cleared.

The fill level information used mainly for performance analysis.

Reading this register alters the internal state of the TMC, and it can be read only if the device is unlocked. Reading this register when the device is locked returns `0x00000000`.

The width of this register in the ETB or ETF configurations is $1+\log_2(\text{MEM\_SIZE})$. In the ETR configuration, the width of this register is 31 bits.

Configurations Present in all configurations.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Figure 3-10 shows the LBUFLEVEL Register bit assignments.

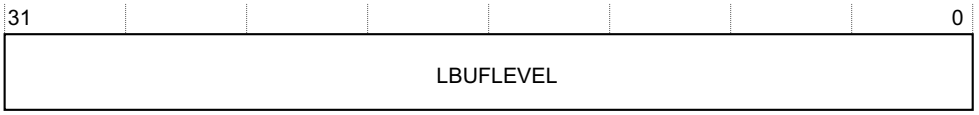


Figure 3-10 LBUFLEVEL Register bit assignments

Table 3-11 shows the LBUFLEVEL Register bit assignments.

Table 3-11 LBUFLEVEL Register bit assignments

Bits	Name	Function
[31:0][a]	LBUFLEVEL	Indicates the maximum fill level of the trace memory in 32-bit words since this register was last read

a. The width of this register is dependent on the TMC configuration.

3.3.11 Current Buffer Fill Level

The CBUFLEVEL Register characteristics are:

Purpose When TraceCaptEn is HIGH, this register indicates the current fill level of the trace memory in 32-bit words. When TraceCaptEn is LOW, reading this register returns a value of `0x00000000`. This register is not available if the TMC is programmed for scatter-gather operation in ETR configuration. In this case, reading this register returns a value of `0x00000000`.

Configurations Present in all configurations.

Figure 3-11 shows the CBUFLEVEL Register bit assignments.

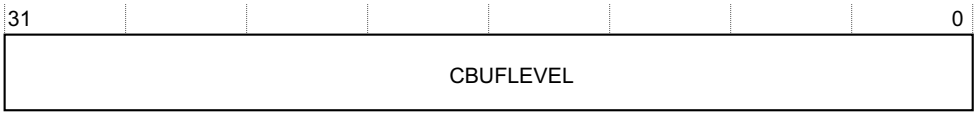


Figure 3-11 CBUFLEVEL Register bit assignments

Table 3-12 shows the CBUFLEVEL Register bit assignments.

Table 3-12 CBUFLEVEL Register bit assignments

Bits	Name	Function
[31:0][a]	CBUFLEVEL	Indicates the current fill level of the trace memory in 32-bit words

a. The width of this register is dependent on the TMC configuration.

3.3.12 Buffer Level Water Mark

The BUFWM Register characteristics are:

Purpose The value programmed into this register indicates the required threshold vacancy level in 32-bit words in the trace memory. When the space in the FIFO is less than or equal to this value, that is, Fill level >= MEM_SIZE - BUFWM, the **FULL** output is pulled HIGH and the Full bit in the STS Register is set.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

This register is used only in Software FIFO and Hardware FIFO modes. In Circular Buffer mode, this functionality can be obtained by programming the RWP to the required vacancy trigger level, so that when the pointer wraps around, the Full bit is set indicating that the vacancy level has fallen below the required level.

The maximum value that can be written into this register is MEM_SIZE - 1, in this case, the Full bit 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.

This register is ignored apply when the TMC is programmed for scatter-gather operation in ETR configuration.

Configurations Present in all configurations.

Figure 3-12 shows the BUFWM Register bit assignments.

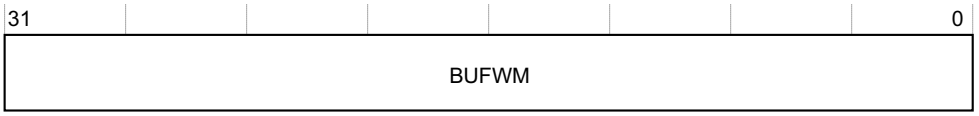


Figure 3-12 BUFWM Register bit assignments

Table 3-13 shows the BUFWM Register bit assignments.

Table 3-13 BUFWM Register bit assignments

Bits	Name	Function
[31:0][a]	BUFWM	Indicates the required threshold vacancy level in 32-bit words in the trace memory

a. The width of this register is dependent on the TMC configuration.

3.3.13 RAM Read Pointer High Register

The RRPHI Register characteristics are:

Purpose In the ETR configuration, memory addresses are 40 bits wide. The RAM Read Pointer High Register sets the upper eight bits of the read pointer that is used together with the contents of the RRP Register to read entries from the trace memory over the APB interface.

See *RAM Read Pointer Register* on page 3-9.

Configurations Present in ETR configuration only.

Figure 3-13 shows the RRPHI Register bit assignments.

Figure 3-13 RRPHI Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-14 shows the RRPHI Register bit assignments.

Table 3-14 RRPHI Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved
[7:0]	RRPHI	Bits [39:32] of the read pointer

3.3.14 RAM Write Pointer High Register

The RWPHI Register characteristics are:

Purpose In the ETR configuration, memory addresses are 40 bits wide. The RAM Write Pointer High Register sets the upper eight bits of the write pointer that is used together with the contents of the RWP Register, to write entries into the trace memory.

See *RAM Write Pointer Register* on page 3-11.

Configurations Present in ETR configuration only.

Figure 3-14 shows the RWPHI Register bit assignments.

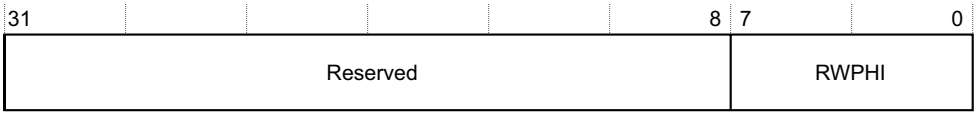


Figure 3-14 RWPHI Register bit assignments

Table 3-15 shows the RWPHI Register bit assignments.

Table 3-15 RWPHI Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved
[7:0]	RWPHI	Bits[39:32] of the write pointer

3.3.15 AXI Control Register

The AXICTL Register characteristics are:

Purpose Controls TMC accesses to system memory through the AXI master interface.

The TMC performs only Data accesses, so **ARPROTM[2]** and **AWPROTM[2]** outputs are LOW for all AXI accesses.

Writing to this register when not in Disabled state results in Unpredictable behavior. In most cases, you can set bits [5:0] of this register to `b111111`.

Configurations Present in ETR configurations only.

Figure 3-15 on page 3-19 shows the AXICTL Register bit assignments.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

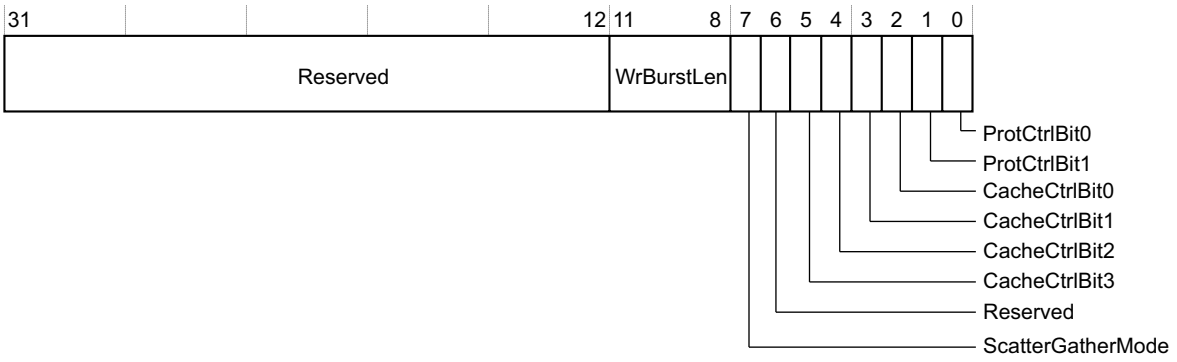


Figure 3-15 AXICTL Register bit assignments

Table 3-16 shows the AXICTL Register bit assignments.

Table 3-16 AXICTL Register bit assignments

Bits	Name	Function
[31:12]	Reserved	Reserved.
[11:8]	WrBurstLen	This field indicates the maximum number of data transfers that can occur within each burst initiated by the TMC on the AXI master interface. The Write burst initiated on the AXI can be of a smaller length that the programmed value in the case when the formatter has stopped because of a stop condition having occurred. Programming this field to a burst length value greater than the write buffer depth results in a burst length that is equal to the write buffer depth. The burst length programmed must be compatible with the trace buffer size and the AXI data width so that the total number of bytes of data transferred in a burst is not greater than the trace buffer size or, if scatter-gather operation is enabled, is not greater than 4KB. Programming an incompatible burst length results in Unpredictable behavior. It is recommended that this value be set to no more than half the write buffer depth. Also, it is recommended that this value be set to enable an AXI burst of at least one frame of trace data. This field is decoded as follows: `0x0` One data transfer per burst. This is the default. `0x1` Maximum of two data transfers per burst. ... `0xF` Maximum of 16 data transfers per burst.
[7]	ScatterGatherMode	This bit indicates whether trace memory is accessed as a single buffer in system memory or as a linked-list based scatter-gather memory. This bit is ignored when in Disabled state. **0** Trace memory is a single contiguous block of system memory. **1** Trace memory is spread over multiple blocks of system memory based on a linked-list mechanism. For more information, see *Scatter-gather* on page 2-16.
[6]	Reserved	Reserved.
[5]	CacheCtrlBit3	This bit controls the value driven on the **ARCACHEM[3]** or **AWCACHEM[3]** signal on the AXI master interface when performing AXI transfers. If CacheCtrlBit1 is LOW, this bit must also be LOW to comply with the AXI protocol. Setting this bit to HIGH when the CacheCtrlBit1 is LOW results in Unpredictable behavior. **0** Do not cache allocate on writes. **1** Cache allocate on writes.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-16 AXICTL Register bit assignments (continued)

Bits	Name	Function
[4]	CacheCtrlBit2	This bit controls the value driven on the **ARCACHEM[2]** or **AWCACHEM[2]** signal on the AXI master interface when performing AXI transfers. If CacheCtrlBit1 is LOW, this bit must also be LOW to comply with the AXI protocol. Setting this bit to HIGH when the CacheCtrlBit1 is LOW results in Unpredictable behavior. **0** Do not cache allocate on reads. **1** Cache allocate on reads.
[3]	CacheCtrlBit1	This bit controls the value driven on the **ARCACHEM[1]** or **AWCACHEM[1]** signal on the AXI master interface when performing AXI transfers. **0** Non-cacheable. **1** Cacheable.
[2]	CacheCtrlBit0	This bit controls the value driven on the **ARCACHEM[0]** or **AWCACHEM[0]** signal on the AXI master interface when performing AXI transfers. **0** Non-bufferable. **1** Bufferable.
[1]	ProtCtrlBit1	This bit controls the value driven on **ARPROTM[1]** or **AWPROTM[1]** on the AXI master interface when performing AXI transfers. **0** Secure access. **1** Non-secure access.
[0]	ProtCtrlBit0	This bit controls the value driven on **ARPROTM[0]** or **AWPROTM[0]** on the AXI master interface when performing AXI transfers. **0** Normal access. **1** Privileged access. A privileged processing mode might have a greater level of access within a system.

3.3.16 Data Buffer Address Low Register

The DBALO Register characteristics are:

Purpose This register, together with the DBAHI Register, enables the TMC to locate the trace buffer in system memory. This register contains the 32 LSBs of the 40-bit address that is used to locate the trace buffer. Modifying this register, other than when in Disabled state results in Unpredictable behavior.

If the ScatterGatherMode bit is LOW in the AXICTL Register, `0x110`, the content of this register is the base address of the trace buffer in system memory.

If the ScatterGatherMode bit is HIGH in the AXICTL Register, `0x110`, the content of this register is the address of first page table entry in the linked list.

Configurations Present in ETR configuration only.

Figure 3-16 shows the DBALO Register bit assignments.

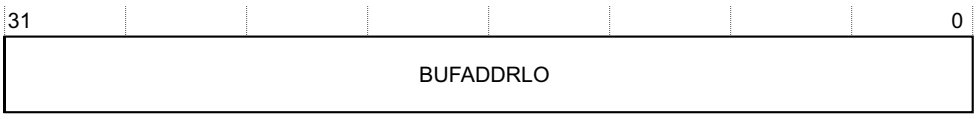


Figure 3-16 DBALO Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-17 shows the DBALO Register bit assignments.

Table 3-17 DBALO Register bit assignments

Bits	Name	Function
[31:0]	BUFADDRLO	Holds the lower 32 bits of the 40-bit address used to locate the trace buffer in system memory

3.3.17 Data Buffer Address High Register

The DBAHI Register characteristics are:

Purpose This register, together with the DBALO Register, enables the TMC to locate the trace buffer in system memory. This register contains the eight MSBs of the 40-bit address that is used to locate the trace buffer. For additional information, see *Data Buffer Address Low Register* on page 3-20.

Configurations Present in ETR configuration only.

Figure 3-17 shows the DBAHI Register bit assignments.

Figure 3-17 DBAHI Register bit assignments

Table 3-18 shows the DBAHI Register bit assignments.

Table 3-18 DBAHI Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved
[7:0]	BUFADDRHI	Holds the upper eight bits of the 40-bit address used to locate the trace buffer in system memory

3.3.18 Formatter and Flush Status Register

The FFSR Register characteristics are:

Purpose Indicates the status of the Formatter and Flush request.

Configurations Present in all configurations.

Figure 3-18 shows the FFSR Register bit assignments.

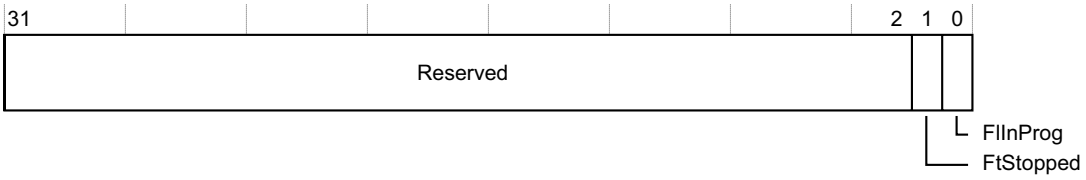


Figure 3-18 FFSR Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-19 shows the FFSR Register bit assignments.

Table 3-19 FFSR Register bit assignments

Bits	Name	Function
[31:2]	Reserved	Reserved.
[1]	FtStopped	This bit behaves in the same way as the FtEmpty bit in the STS Register, `0x00C`. The FtStopped bit is deprecated and is present in this register to support backwards-compatibility with earlier versions of the ETB.
[0]	FlInProg	This bit indicates whether the TMC is currently processing a flush on the ATB slave port. This bit reflects the status of the **AFVALIDS** output. In the ETB or ETR configurations, the flush initiation is controlled by the flush-control bits in the FFCR Register. In the ETF configuration, the flush request could additionally be from the ATB master port. **0** No flush activity in progress on the ATB slave interface. **1** Flush in progress on the ATB slave interface.

3.3.19 Formatter and Flush Control Register

The FFCR Register characteristics are:

Purpose Controls the generation of the stop, trigger, and flush events.

Multiple flush generating conditions can be enabled together. However, if a second or third flush event is generated, then the current flush completes before the next flush is serviced.

Multiple trigger indication conditions can be enabled simultaneously, which can cause the appearance of multiple triggers in the trace stream.

Note

To perform a stop on flush completion through a manually-generated flush request, two write operations to the register are required:

- one to enable the stop event, if it is not already enabled
- one to generate the manual flush.

Configurations Present in all configurations.

Figure 3-19 on page 3-23 shows the FFCR Register bit assignments.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

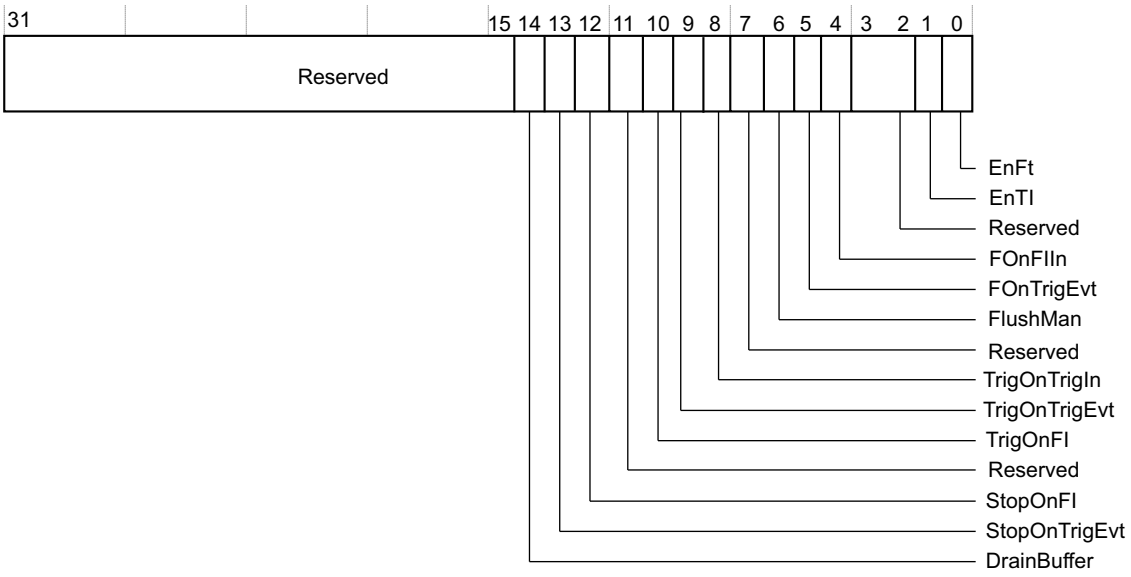


Figure 3-19 FFCR Register bit assignments

Table 3-20 shows the FFCR Register bit assignments.

Table 3-20 FFCR Register bit assignments

Bits	Name	Function
[31:15]	Reserved	Reserved.
[14]	DrainBuffer	In ETF configuration, this bit is used to enable draining of the trace data through the ATB master interface after the formatter has stopped. This is useful in Circular Buffer mode to capture trace data into trace memory and then to drain the captured trace through the ATB master interface. Writing a one to this bit when in Stopped state starts the drain of the contents of the trace buffer through the ATB Master interface. This bit always reads as zero. The TMCReady bit, STS Register, `0x00C`, goes LOW while the drain is in progress. This bit is functional only when the TMC is in Circular Buffer mode and formatting is enabled, EnFt bit in FFCR Register is set. 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 Circular Buffer mode, all of the captured trace must be retrieved from the trace memory through the same mechanism, either read all trace data out through RRD reads, or drain all trace data by setting the DrainBuffer bit. Setting the DrainBuffer bit after some of the captured trace has been read out through RRD results in Unpredictable behavior. **0** Trace data is not drained through ATB master interface. **1** Trace data is drained through ATB master interface.
[13]	StopOnTrigEvt	If this bit is set, the formatter is stopped when a Trigger Event has been observed. For more information on Trigger Events, see *Trigger, flush and stop events* on page 2-13. This bit is cleared on reset. Enabling the TMC in Software FIFO mode, or Hardware FIFO mode, with this bit set results in Unpredictable behavior. **0** Trace capture is not stopped when a Trigger Event is observed. **1** Trace capture is stopped when a Trigger Event is observed.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-20 FFCR Register bit assignments (continued)

Bits	Name	Function
[12]	StopOnFl	If this bit is set, the formatter is stopped on completion of a flush operation. For more information on Trigger Events, see *Trigger, flush and stop events* on page 2-13. 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 programmed in this bit. **0** Trace capture is not stopped when FLUSH is completed. **1** Trace capture is stopped when FLUSH is completed.
[11]	Reserved	Reserved.
[10]	TrigOnFl	If this bit is set, a trigger is indicated on the trace stream when a flush completes. For more information, see *Trigger, flush and stop events* on page 2-13. If EnFt and EnTI are both clear, this bit is ignored and no trigger is inserted into the trace stream. 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 trigger indication on the trace stream regardless of the value programmed in this bit. **0** A trigger is not indicated on the trace stream when a flush completes. **1** A trigger is indicated on the trace stream when a flush completes.
[9]	TrigOnTrigEvt	If this bit is set, a trigger is indicated on the output trace stream when a Trigger Event occurs. For more information, see *Trigger, flush and stop events* on page 2-13. If EnFt and EnTI are both clear, this bit is ignored and no trigger is inserted into the trace stream. This bit is not supported in Software FIFO mode or Hardware FIFO mode. **0** A trigger is not indicated on the trace stream when a Trigger Event occurs. **1** A trigger is indicated on the trace stream when a Trigger Event occurs.
[8]	TrigOnTrigIn	If this bit is set, a trigger is indicated on the trace stream when a rising edge is detected on the **TRIGIN** input. For more information, see *Trigger, flush and stop events* on page 2-13. If EnFt and EnTI are both clear, this bit is ignored and no trigger is inserted into the trace stream. **0** A trigger is not indicated on the trace stream when a rising edge is detected on the **TRIGIN** input. **1** A trigger is indicated on the trace stream when a rising edge is detected on the **TRIGIN** input.
[7]	Reserved	Reserved.
[6]	FlushMan	Manually generate a flush of the system. Setting this bit causes a flush to be generated. If TraceCaptEn bit in CTL Register is 0, then writes to this bit are ignored. This bit is cleared automatically when a flush completes. For more information, see *Trigger, flush and stop events* on page 2-13. This bit is clear on reset. **0** Manual Flush is not initiated. **1** Manual Flush is initiated.
[5]	FOnTrigEvt	Setting this bit generates a flush when a Trigger Event occurs. If StopOnTrigEvt is set, this bit is ignored. For more information, see *Trigger, flush and stop events* on page 2-13. This bit is clear on reset. This bit is not supported in Software FIFO mode or Hardware FIFO mode. **0** Flush-on-trigger-event disabled. **1** Flush-on-trigger-event enabled.
[4]	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. This bit is clear on reset. **0** Flush-on-FLUSHIN feature is disabled. **1** Flush-on-FLUSHIN feature is enabled.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-20 FFCR Register bit assignments (continued)

Bits	Name	Function
[3:2]	Reserved	Reserved
[1]	EnTI	Setting this bit enables the insertion of triggers in the formatted trace stream. A trigger is indicated by inserting one byte of data `8'h00` with **ATID** `7'h7D` in the trace stream. Trigger indication on the trace stream is additionally controlled by the register bits TrigOnFl, TrigOnTrigEvt, and TrigOnTrigIn in the FFCR Register, `0x304`. This bit can only be changed when TMCReady is HIGH, and TraceCaptEn is LOW. This bit takes effect only when the EnFt register bit in this register is set. If EnTI bit is set to HIGH when EnFt is LOW, it results in formatting being enabled. This bit is clear on reset. **0** Triggers are not embedded into the formatted trace stream. **1** Embed triggers into formatted trace stream.
[0]	EnFt	If this bit is set, formatting is enabled. This bit is clear on reset. This bit is ignored when in Disabled state. If this bit is clear, formatting is disabled. Incoming trace data is assumed to be from a single trace source. If multiple ATIDs 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. Disabling of formatting is supported only in Circular Buffer mode. Features in the TMC such as the FIFO modes and the DrainBuffer bit that are not part of the earlier versions of the ETB do not support disabling of formatting. If EnTI bit is set to HIGH when EnFt is LOW, it results in formatting being enabled. If the TMC is enabled in a mode other than Circular Buffer mode with EnFt LOW, it results in formatting being enabled. **0** Formatting disabled. **1** Formatting enabled.

3.3.20 Periodic Synchronization Counter Register

The PSCR Register characteristics are:

Purpose Determines the reload value of the Periodic Synchronization Counter. This enables the frequency of synchronization information to be optimized to the trace capture buffer size.

This counter is enabled only when the TraceCaptEn bit in the Control Register, CTL, `0x020`, is set. Writing to this register, other than when in Disabled state, results in Unpredictable behavior.

Configurations Present in all configurations.

Figure 3-20 shows the PSCR Register bit assignments.

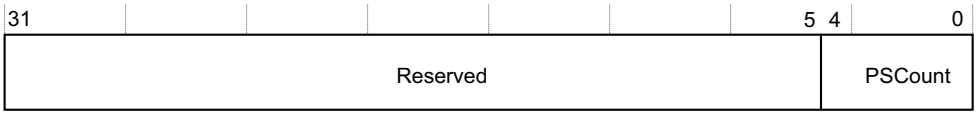


Figure 3-20 PSCR Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-21 shows the PSCR Register bit assignments.

Table 3-21 PSCR Register bit assignments

Bits	Name	Function
[31:5]	Reserved	Reserved.
[4:0]	PSCount	Determines the reload value of the Synchronization Counter. The reload value takes effect the next time the counter reaches zero. Reads from this register return the reload value programmed into this register. This register is set to `0xA` on reset, corresponding to a synchronization period of 1024 bytes. `0x0` Synchronization disabled. `0x1-0x6` Reserved. `0x7-0x1B` Synchronization period is $2^{PSCount\ bytes}$. For example, a value of `0x7` gives a synchronization period of 128 bytes. `0x1C-0x1F` Reserved.

3.3.21 Integration Test ATB Master Data Register 0

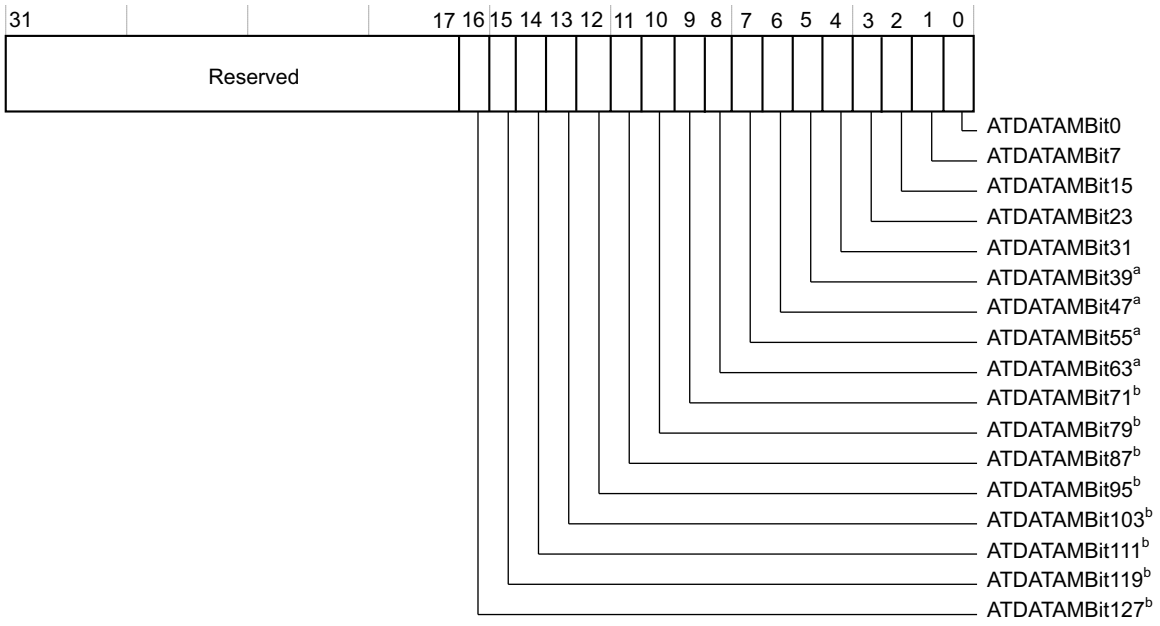
The ITATBMDATA0 Register characteristics are:

Purpose Enables control of the ATDATAM output of the TMC.

Writing to this register other than when in Disabled state and in integration mode results in Unpredictable behavior.

Configurations Present in ETF configuration only.

Figure 3-21 shows the ITATBMDATA0 Register bit assignments.



[a] - Available in 64-bit and 128-bit configurations only.
[b] - Available in 128-bit configurations only.

Figure 3-21 ITATBMDATA0 Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-22 shows the ITATBMDATA0 Register bit assignments.

Table 3-22 ITATBMDATA0 Register bit assignments

Bits	Name	Function
[31:17]	Reserved	Reserved
[16]	ATDATAMBit127	Control the value of **ATDATAM[127]** output of TMC. Present in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[15]	ATDATAMBit119	Control the value of **ATDATAM[119]** output of TMC. Present in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[14]	ATDATAMBit111	Control the value of **ATDATAM[111]** output of TMC. Present in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[13]	ATDATAMBit103	Control the value of **ATDATAM[103]** output of TMC. Present in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[12]	ATDATAMBit95	Control the value of **ATDATAM[95]** output of TMC. Present in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[11]	ATDATAMBit87	Control the value of **ATDATAM[87]** output of TMC. Present in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[10]	ATDATAMBit79	Control the value of **ATDATAM[79]** output of TMC. Available in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[9]	ATDATAMBit71	Control the value of **ATDATAM[71]** output of TMC. Present in 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[8]	ATDATAMBit63	Control the value of **ATDATAM[63]** output of TMC. Present in 64-bit and 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[7]	ATDATAMBit55	Control the value of **ATDATAM[55]** output of TMC. Present in 64-bit and 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-22 ITATBMDATA0 Register bit assignments (continued)

Bits	Name	Function
[6]	ATDATAMBit47	Control the value of **ATDATAM[47]** output of TMC. Present in 64-bit and 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[5]	ATDATAMBit39	Control the value of **ATDATAM[39]** output of TMC. Present in 64-bit and 128-bit configurations only. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[4]	ATDATAMBit31	Control the value of **ATDATAM[31]** output of TMC. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[3]	ATDATAMBit23	Control the value of **ATDATAM[23]** output of TMC. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[2]	ATDATAMBit15	Control the value of **ATDATAM[15]** output of TMC. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[1]	ATDATAMBit7	Control the value of **ATDATAM[7]** output of TMC. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.
[0]	ATDATAMBit0	Control the value of **ATDATAM[0]** output of TMC. **0** Drive logic 0 on this output. **1** Drive logic 1 on this output.

3.3.22 Integration Test ATB Master Interface Control 2 Register

The ITATBMCTR2 Register characteristics are:

Purpose Captures the values of the **SYNCREQM**, **AFVALIDM**, and **ATREADYM** inputs to the TMC.

Configurations Present in ETF configuration only.

Figure 3-22 shows the ITATBMCTR2 Register bit assignments.

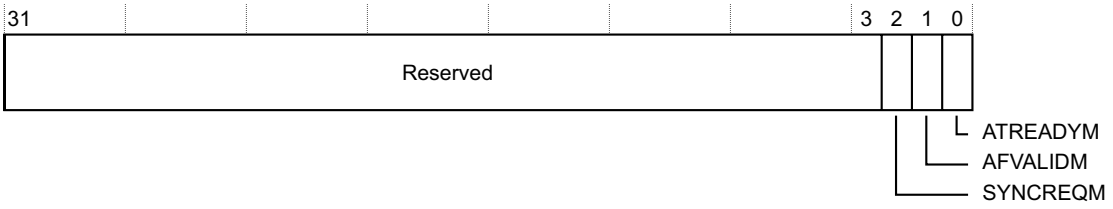


Figure 3-22 ITATBMCTR2 Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-23 shows the ITATBMCTR2 Register bit assignments.

Table 3-23 ITATBMCTR2 Register bit assignments

Bits	Name	Function
[31:3]	Reserved	Reserved.
[2]	SYNCREQM	Read the value of **SYNCREQM** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.
[1]	AFVALIDM	Read the value of **AFVALIDM** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.
[0]	ATREADYM	Read the value of **ATREADYM** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.

3.3.23 Integration Test ATB Master Control Register 1

The ITATBMCTR1 Register characteristics are:

Purpose Enables control of the **ATIDM** outputs of the TMC.

Writing to this register other than when in Disabled state and in integration mode results in Unpredictable behavior.

Configurations Present in ETF configurations only.

Figure 3-23 shows the ITATBMCTR1 Register bit assignments.

Figure 3-23 ITATBMCTR1 Register bit assignments

Table 3-24 shows the ITATBMCTR1 Register bit assignments.

Table 3-24 ITATBMCTR1 Register bit assignments

Bits	Name	Function
[31:7]	Reserved	Reserved.
[6:0]	ATIDM	Control the value of **ATIDM** output from TMC. The value written to this field is driven on the ATIDM output of the TMC.

3.3.24 Integration Test ATB Master Interface Control 0 Register

The ITATBMCTR0 Register characteristics are:

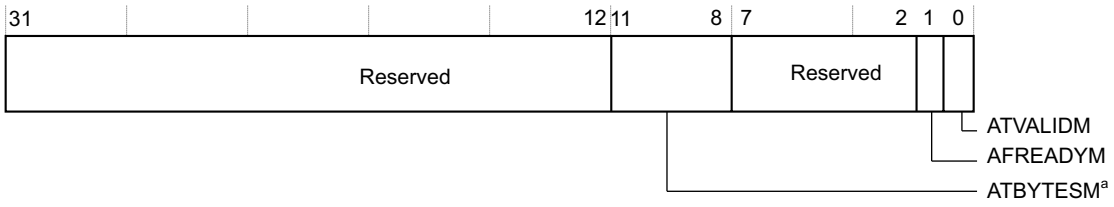
Purpose Enables control of the **ATBYTESM**, **AFREADYM**, and **ATVALIDM** outputs of the TMC.

Writing to this register other than when in Disabled state and in integration mode results in Unpredictable behavior.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Configurations Present in ETR configuration only.

Figure 3-24 shows the ITATBMCTR0 Register bit assignments.



[a] - The width of the ATBYTES field depends on the configuration:
- for 32-bit configurations it is [9:8]
- for 64-bit configurations it is [10:8]
- for 128-bit configurations it is [11:8].

Figure 3-24 ITATBMCTR0 Register bit assignments

Table 3-25 shows the ITATBMCTR0 Register bit assignments.

Table 3-25 ITATBMCTR0 Register bit assignments

Bits	Name	Function
[31:12]	Reserved	Reserved.
[11:8]	ATBYTESM[a]	Control the value of **ATBYTESM** output from TMC. The value written to this field is driven on the **ATBYTESM** output of the TMC.
[7:2]	Reserved	Reserved.
[1]	AFREADYM	Set the value of **AFREADYM** output. **0** Drive logic 0 on **AFREADYM** output. **1** Drive logic 1 on **AFREADYM** output.
[0]	ATVALIDM	Set the value of **ATVALIDM** output. **0** Drive logic 0 on **ATVALIDM** output. **1** Drive logic 1 on **ATVALIDM** output.

a. The width of the ATBYTES field depends on the configuration:
[9:8] For 32-bit configurations.
[10:8] For 64-bit configurations.
[11:8] For 128-bit configurations.

3.3.25 Integration Test Miscellaneous Output Register 0

The ITMISCOP0 Register characteristics are:

Purpose Controls the values of some outputs from the TMC.

Writing to this register other than when in Disabled state and in integration mode results in Unpredictable behavior.

Configurations Present in all configurations.

Figure 3-25 on page 3-31 shows the ITMISCOP0 Register bit assignments.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

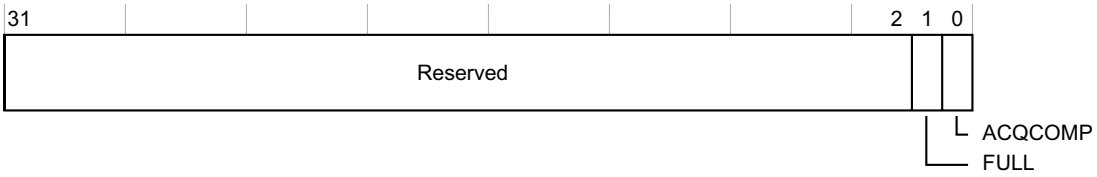


Figure 3-25 ITMISCOP0 Register bit assignments

Table 3-26 shows the ITMISCOP0 Register bit assignments.

Table 3-26 ITMISCOP0 Register bit assignments

Bits	Name	Function
[31:2]	Reserved	Reserved.
[1]	FULL	Set the value of the **FULL** output. **0** Drive logic 0 on **FULL** output. **1** Drive logic 1 on **FULL** output.
[0]	ACQCOMP	Set the value of the **ACQCOMP** output. **0** Drive logic 0 on **ACQCOMP** output. **1** Drive logic 1 on **ACQCOMP** output.

3.3.26 Integration Test Trigger In and Flush In Register

The ITTRFLIN Register characteristics are:

Purpose Returns the values of the **FLUSHIN** and **TRIGIN** inputs to the TMC.

Configurations Present in all configurations.

Figure 3-26 shows the ITTRFLIN Register bit assignments.

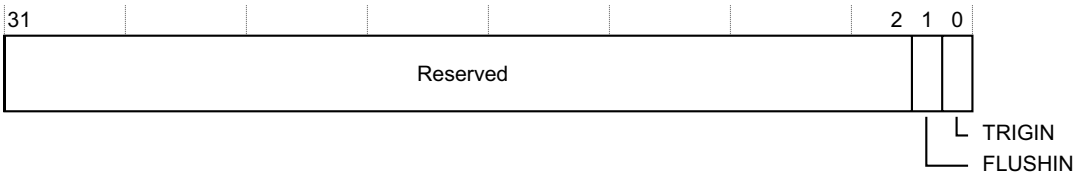


Figure 3-26 ITTRFLIN Register bit assignments

Table 3-27 shows the ITTRFLIN Register bit assignments.

Table 3-27 ITTRFLIN Register bit assignments

Bits	Name	Function
[31:2]	Reserved	Reserved.
[1]	FLUSHIN	Read the value of the **FLUSHIN** input. **0** Pin is at logic 0. **1** Pin is at logic 1.
[0]	TRIGIN	Read the value of the **TRIGIN** input. **0** Pin is at logic 0. **1** Pin is at logic 1.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

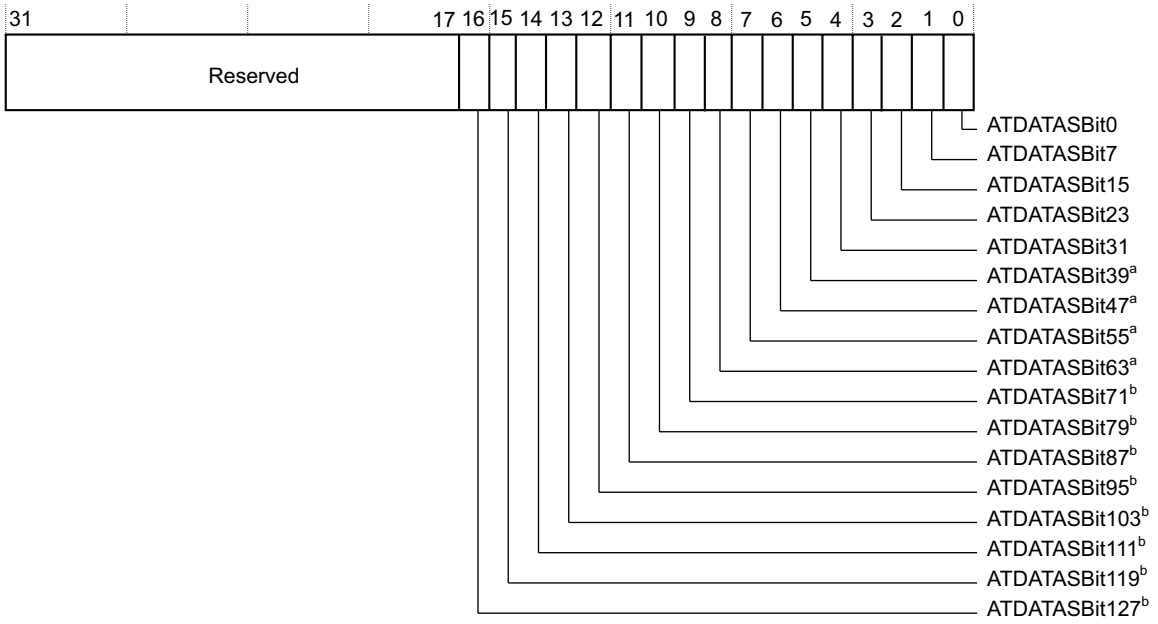
3.3.27 Integration Test ATB Data Register 0

The ITATBDATA0 Register characteristics are:

Purpose Returns the values on the **ATDATAS** input to the TMC.

Configurations Present in all configurations.

Figure 3-27 shows the ITATBDATA0 Register bit assignments.



[a] - Available in 64-bit and 128-bit configurations only.
[b] - Available in 128-bit configurations only.

Figure 3-27 ITATBDATA0 Register bit assignments

Table 3-28 shows the ITATBDATA0 Register bit assignments.

Table 3-28 ITATBDATA0 Register bit assignments

Bits	Name	Function
[31:17]	Reserved	Reserved.
[16]	ATDATASBit127	Read the value of **ATDATAS[127]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[15]	ATDATASBit119	Read the value of **ATDATAS[119]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-28 ITATBDATA0 Register bit assignments (continued)

Bits	Name	Function
[14]	ATDATASBit111	Read the value of **ATDATAS[111]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[13]	ATDATASBit103	Read the value of **ATDATAS[103]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[12]	ATDATASBit95	Read the value of **ATDATAS[95]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[11]	ATDATASBit87	Read the value of **ATDATAS[87]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[10]	ATDATASBit79	Read the value of **ATDATAS[79]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[9]	ATDATASBit71	Read the value of **ATDATAS[71]** input to TMC. Present in 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[8]	ATDATASBit63	Read the value of **ATDATAS[63]** input to TMC. Present in 64-bit and 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[7]	ATDATASBit55	Read the value of **ATDATAS[55]** input to TMC. Present in 64-bit and 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[6]	ATDATASBit47	Read the value of **ATDATAS[47]** input to TMC. Present in 64-bit and 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[5]	ATDATASBit39	Read the value of **ATDATAS[39]** input to TMC. Available in 64-bit and 128-bit configurations only. **0** Pin is at logic 0. **1** Pin is at logic 1.
[4]	ATDATASBit31	Read the value of **ATDATAS[31]** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-28 ITATBDATA0 Register bit assignments (continued)

Bits	Name	Function
[3]	ATDATASBit23	Read the value of **ATDATAS[23]** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.
[2]	ATDATASBit15	Read the value of **ATDATAS[15]** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.
[1]	ATDATASBit7	Read the value of **ATDATAS[7]** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.
[0]	ATDATASBit0	Read the value of **ATDATAS[0]** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.

3.3.28 Integration Test ATB Control 2 Register

The ITATBCTR2 Register characteristics are:

Purpose Enables control of the **ATREADYS** and **AFVALIDS** outputs of the TMC.

Writing to this register other than when in Disabled state and in integration mode results in Unpredictable behavior.

Configurations Present in all configurations.

Figure 3-28 shows the ITATBCTR2 Register bit assignments.

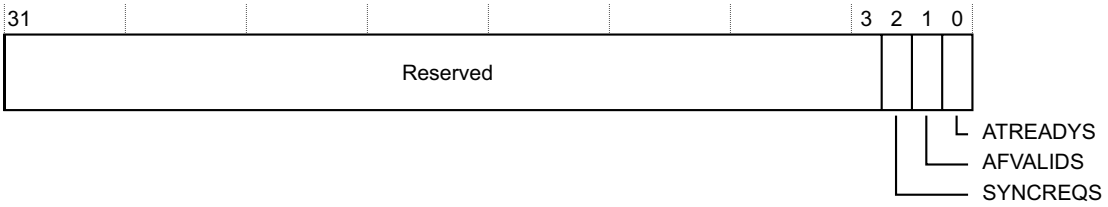


Figure 3-28 ITATBCTR2 Register bit assignments

Table 3-29 shows the ITATBCTR2 Register bit assignments.

Table 3-29 ITATBCTR2 Register bit assignments

Bits	Name	Function
[31:3]	Reserved	Reserved.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-29 ITATBCTR2 Register bit assignments (continued)

Bits	Name	Function
[2]	SYNCREQS	Set the value of **SYNCREQS** output. **0** Drive logic 0 on **SYNCREQS** output. **1** Drive logic 1 on **SYNCREQS** output.
[1]	AFVALIDS	Set the value of **AFVALIDS** output. **0** Drive logic 0 on **AFVALIDS** output. **1** Drive logic 1 on **AFVALIDS** output.
[0]	ATREADYS	Set the value of **ATREADYS** output. **0** Drive logic 0 on **ATREADYS** output. **1** Drive logic 1 on **ATREADYS** output.

3.3.29 Integration Test ATB Control 1 Register

The ITATBCTR1 Register characteristics are:

Purpose Captures the value of the **ATIDS** input to the TMC.

Configurations Present in all configurations.

Figure 3-29 shows the ITATBCTR1 Register bit assignments.

Figure 3-29 ITATBCTR1 Register bit assignments

Table 3-30 shows the ITATBCTR1 Register bit assignments.

Table 3-30 ITATBCTR1 Register bit assignments

Bits	Name	Function
[31:7]	Reserved	Reserved
[6:0]	ATIDS	Reads the value of **ATIDS** input to TMC

3.3.30 Integration Test ATB Control 0 Register

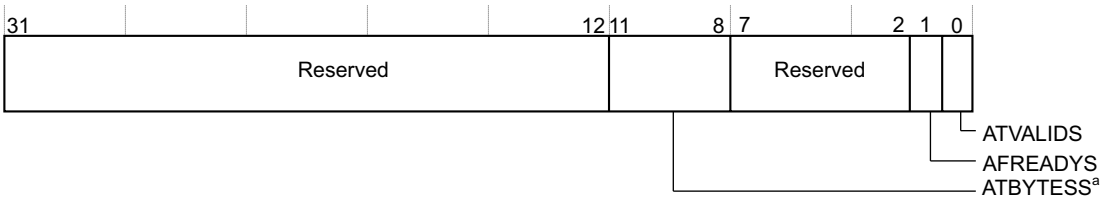
The ITATBCTR0 Register characteristics are:

Purpose Captures the values of the **ATVALIDS**, **AFREADYS**, and **ATBYTESS** inputs to the TMC.

Configurations Present in all configurations.

Figure 3-30 on page 3-36 shows the ITATBCTR0 Register bit assignments.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential



[a] - The width of the ATBYTES field depends on the configuration:
- for 32-bit configurations it is [9:8]
- for 64-bit configurations it is [10:8]
- for 128-bit configurations it is [11:8].

Figure 3-30 ITATBCTR0 Register bit assignments

Table 3-31 shows the ITATBCTR0 Register bit assignments.

Table 3-31 ITATBCTR0 Register bit assignments

Bits	Name	Function
[31:12]	Reserved	Reserved.
[11:8]	ATBYTESS[a]	Read the value of **ATBYTESS** input to TMC.
[7:2]	Reserved	Reserved.
[1]	AFREADYS	Read the value of **AFREADYS** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.
[0]	ATVALIDS	Read the value of **ATVALIDS** input to TMC. **0** Pin is at logic 0. **1** Pin is at logic 1.

a. The width of the ATBYTES field depends on the configuration.
[9:8] For 32-bit configurations.
[10:8] For 64-bit configurations.
[11:8] For 128-bit configurations.

3.3.31 Integration Mode Control Register

The ITCTRL Register characteristics are:

Purpose Enables topology detection, and integration testing. For more information, see the *CoreSight Architecture Specification*. This register enables the component to switch from a functional mode, the default behavior, to integration mode where the inputs and outputs of the component can be directly controlled for the purpose of integration testing and topology solving.

Note

When a device has been in integration mode, it might not function with the original behavior. After performing integration or topology detection, you must reset the system to ensure correct behavior of CoreSight and other connected system components that are affected by the integration or topology detection.

Writing to this register other than when in Disabled state results in Unpredictable behavior.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Configurations Present in all configurations.

Figure 3-31 shows the ITCTRL Register bit assignments.

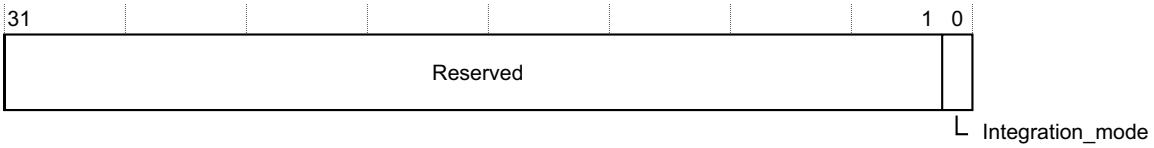


Figure 3-31 ITCTRL Register bit assignments

Table 3-32 shows the ITCTRL Register bit assignments.

Table 3-32 ITCTRL Register bit assignments

Bits	Name	Function
[31:1]	Reserved	Reserved.
[0]	Integration_mode	Enables the component to switch from functional mode to integration mode or back. **0** Disable integration mode. **1** Enable integration mode.

3.3.32 Claim Tag Set Register

The CLAIMSET Register characteristics are:

Purpose This is used in conjunction with the Claim Tag Clear Register, CLAIMCLR. This register forms one half of the Claim Tag value. This location enables individual bits to be set, written and returns the number of bits that can be set, read.

Configurations Present in all configurations.

Figure 3-32 shows the CLAIMSET Register bit assignments.

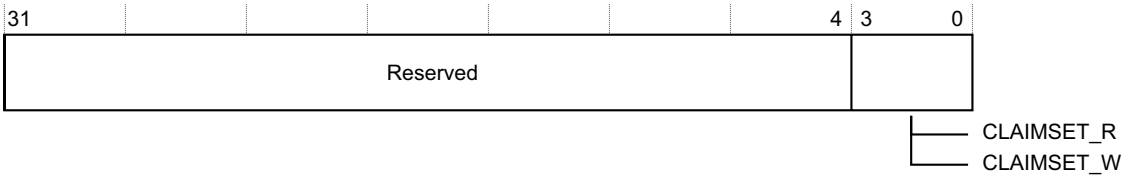


Figure 3-32 CLAIMSET Register bit assignments

Table 3-33 shows the CLAIMSET Register bit assignments.

Table 3-33 CLAIMSET Register bit assignments

Bits	Name	Function	
[31:4]	Reserved	Reserved.	
[3:0]	CLAIMSET_W	On writes:	
		0	No effect.
		1	Set this bit in the claim tag.
	CLAIMSET_R	On reads:	
		`0xF`	Four bits are present within the claim tag field.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.33 Claim Tag Clear Register

The CLAIMCLR Register characteristics are:

Purpose This register is used in conjunction with the Claim Tag Set Register, CLAIMSET. This register forms one half of the Claim Tag value. This location enables individual bits to be cleared, write, and returns the current Claim Tag value, read.

Configurations Present in all configuration.

Figure 3-33 shows the CLAIMCLR Register bit assignments.

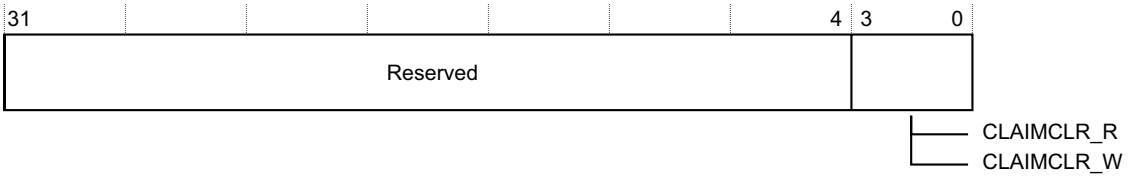


Figure 3-33 CLAIMCLR Register bit assignments

Table 3-34 shows the CLAIMCLR Register bit assignments.

Table 3-34 CLAIMCLR Register bit assignments

Bits	Name	Function	
[31:4]	Reserved	Reserved.	
[3:0]	CLAIMCLR_W	On writes:	
		0	No effect.
		1	Clear this bit in the claim tag.
	CLAIMCLR_R	On reads, returns the current setting of the claim tag.	

3.3.34 Lock Access Register

The LAR Register characteristics are:

Purpose Enables write access to device registers.

External accesses from a debugger, **PADDRDBG31** = 1, are not subject to the Lock registers. A debugger does not have to unlock the component to write and modify the registers in the component.

Configurations Present in all configurations.

Figure 3-34 shows the LAR Register bit assignments.

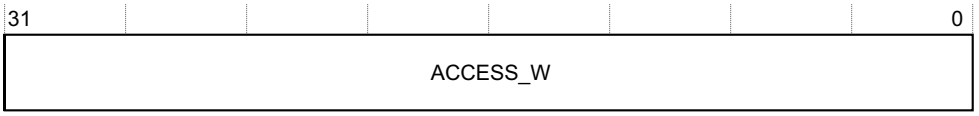


Figure 3-34 LAR Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-35 shows the LAR Register bit assignments.

Table 3-35 LAR Register bit assignments

Bits	Name	Function
[31:0]	ACCESS_W	A write of 0xC5ACCE55 enables additional write access to this device. A write of any value other than 0xC5ACCE55 can have the affect of removing write access.

3.3.35 Lock Status Register

The LSR Register characteristics are:

Purpose Indicates the status of the lock control mechanism. This lock prevents accidental writes by code under debug. When locked, write access is blocked to all registers except the Lock Access Register.

External accesses from a debugger, **PADDRDBG31** = 1, are not subject to the Lock registers. This register reads as 0 when read from an external debugger, **PADDRDBG31** = 1.

Configurations Present in all configurations.

Figure 3-35 shows the LSR Register bit assignments.

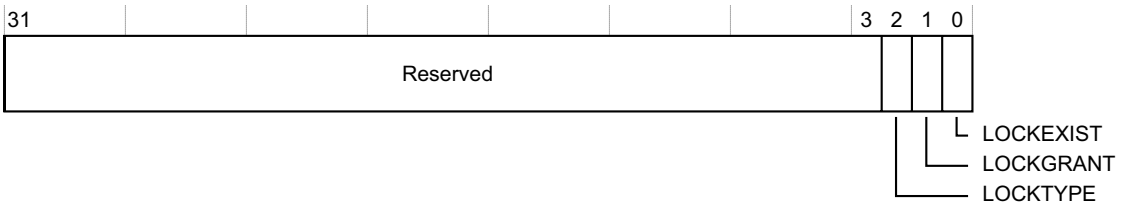


Figure 3-35 LSR Register bit assignments

Table 3-36 shows the LSR Register bit assignments.

Table 3-36 LSR Register bit assignments

Bits	Name	Function
[31:3]	Reserved	Reserved.
[2]	LOCKTYPE	Indicates whether the Lock Access Register (0xFB0) is implemented as 8-bit or 32-bit. 0x0 This component implements a 32-bit Lock Access Register.
[1]	LOCKGRANT	Returns the current status of the lock. This bit reads as zero when read from an external debugger, **PADDRDBG31** = 1, because external debugger accesses are not subject to Lock Registers. 0x0 Write access is permitted to this device. 0x1 Write access to the component is blocked. All writes to control registers are ignored. Reads are permitted.
[0]	LOCKEXIST	Indicates that a lock control mechanism exists for this device. This bit reads as 0 when read from an external debugger, **PADDRDBG31** = 1, because external debugger accesses are not subject to lock registers. 0x0 No lock control mechanism exists, writes to the Lock Access Register, 0xFB0, are ignored. 0x1 Lock control mechanism is present.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.36 Authentication Status Register

The AUTHSTATUS Register characteristics are:

Purpose Reports what functionality is currently permitted by the authentication interface.

Configurations Present in all configurations.

Figure 3-36 shows the AUTHSTATUS Register bit assignments.

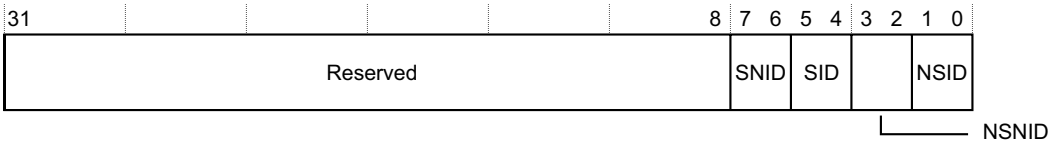


Figure 3-36 AUTHSTATUS Register bit assignments

Table 3-37 shows the AUTHSTATUS Register bit assignments.

Table 3-37 AUTHSTATUS Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:6]	SNID	Indicates the security level for secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere.
[5:4]	SID	Indicates the security level for secure invasive debug. `0x0` Functionality not implemented. This return value occurs when the TMC is configured as an ETB or ETF. `0x2` Functionality disabled. This return value occurs when the TMC is configured as an ETR, and **SPIDEN** is LOW or **DBGEN** is LOW. `0x3` Functionality enabled. This return value occurs when the TMC is configured as an ETR, and **SPIDEN** and **DBGEN** are HIGH.
[3:2]	NSNID	Indicates the security level for non-secure non-invasive debug. `0x0` Functionality not implemented or controlled elsewhere.
[1:0]	NSID	Indicates the security level for non-secure invasive debug. `0x0` Functionality not implemented. This return value occurs when the TMC is configured as an ETB or ETF. `0x2` Functionality disabled. This return value occurs when the TMC is configured as an ETR, and **DBGEN** is LOW. `0x3` Functionality enabled. This return value occurs when the TMC is configured as an ETR, and **DBGEN** is HIGH.

3.3.37 Device Configuration Register

The DEVID Register characteristics are:

Purpose Indicates the capabilities of the CoreSight TMC.

Configurations Present in all configurations.

Figure 3-37 on page 3-41 shows the DEVID Register bit assignments.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

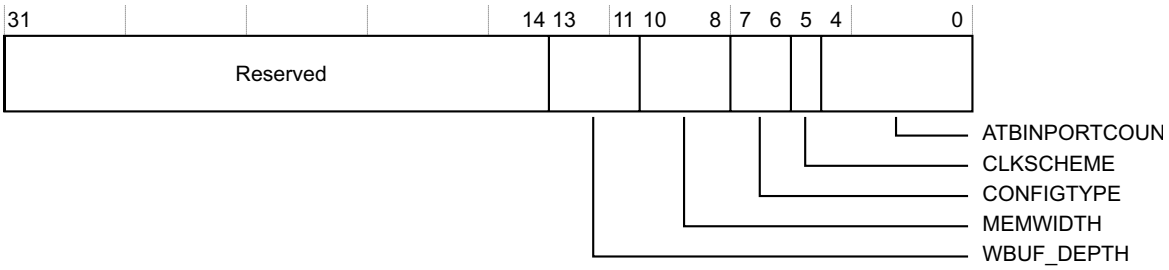


Figure 3-37 DEVID Register bit assignments

Table 3-38 shows the DEVID Register bit assignments.

Table 3-38 DEVID Register bit assignments

Bits	Name	Function
[31:14]	Reserved	Reserved.
[13:11]	WBUF_DEPTH	In the ETR configuration, this value indicates, in powers of two, the number of entries in the Write buffer. Each entry is of size ATB_DATA_WIDTH. `0x2` Depth of the Write buffer is 4 entries. `0x3` Depth of the Write buffer is 8 entries. `0x4` Depth of the Write buffer is 16 entries. `0x5` Depth of the Write buffer is 32 entries.
[10:8]	MEMWIDTH	This value indicates the width of the Memory interface databus. For the ETB and ETF configurations, you have chose this value that is twice the ATB datawidth. In these configurations, the default value of MEMWIDTH is 64 bits, corresponding to the default ATB datawidth of 32 bits. For the ETR configuration, the MEMWIDTH is the same as the ATB datawidth, the default value being 32 bits. `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]	CONFIGTYPE	This value indicates TMC configuration types. `0x0` ETB. `0x1` ETR. `0x2` ETF.
[5]	CLKSCHEME	This value indicates the TMC RAM clocking scheme used. That is, whether the TMC RAM operates synchronously or asynchronously to **CLK**. `0x0` The TMC RAM operates synchronously to **CLK**.
[4:0]	ATBINPORTCOUNT	This value indicates the type or number of ATB multiplexing that is available on the input ATB. Currently, only `0x00` is supported. This value is used to assist topology detection of the ATB structure. `0x0` Currently, only `0x00` is supported, and no multiplexing is available.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.38 Device Type Identifier Register

The DEVTYPE Register characteristics are:

Purpose Provides a debugger with information about the component when the Part Number field is not recognized. The debugger can then report this information.

Configurations Present in all configurations.

Figure 3-38 shows the DEVTYPE Register bit assignments.

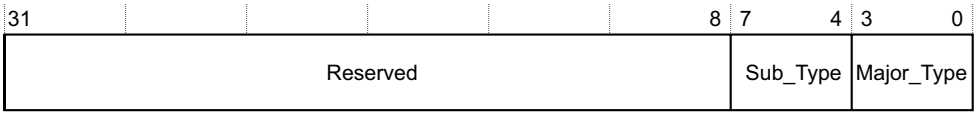


Figure 3-38 DEVTYPE Register bit assignments

Table 3-39 shows the DEVTYPE Register bit assignments.

Table 3-39 DEVTYPE Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:4]	Sub_Type	The following sub-classifications exist within the major category: 0x2 When configured as an ETB or ETR, this component captures the trace data into RAM that can be drained through APB. 0x3 When configured as an ETF, this component captures the trace data from the ATB slave interface into RAM that can be drained through the ATB master interface.
[3:0]	Major_Type	The following are the major classification groups for this debug or trace component: 0x1 When configured as an ETB or ETR, this component is a trace sink. 0x2 When configured as an ETF, this component is a trace link because it has an ATB master interface through which trace data can be drained out in Hardware FIFO mode.

3.3.39 Peripheral ID4 Register

The PERIPHID4 Register characteristics are:

Purpose Part of the set of Peripheral Identification registers. Contains part of the designer identity and the memory footprint indicator.

Configurations Present in all configurations.

Figure 3-39 shows the PERIPHID4 Register bit assignments.

Figure 3-39 PERIPHID4 Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-40 shows the PERIPHID4 Register bit assignments.

Table 3-40 PERIPHID4 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:4]	FourKB_Count	This is a 4-bit value that indicates the total contiguous size of the memory window used by this component in powers of two from the standard 4KB. `0x0` Indicates that the device only occupies 4KB of memory.
[3:0]	JEP106_cont	JEDEC continuation code indicating the designer of the component and the identity code. `0x4` Indicates that ARM's JEDEC identity code is on the fifth bank.

3.3.40 Peripheral ID5 Register

The PERIPHID5 Register characteristics are:

Purpose Reserved.

Configurations Present in all configurations.

Figure 3-40 shows the PERIPHID5 Register bit assignments.

Figure 3-40 PERIPHID5 Register bit assignments

Table 3-41 shows the PERIPHID5 Register bit assignments.

Table 3-41 PERIPHID5 Register bit assignments

Bits	Name	Function
[31:0]	Reserved	Reserved

3.3.41 Peripheral ID6 Register

The PERIPHID6 Register characteristics are:

Purpose Reserved.

Configurations Present in all configurations.

Figure 3-41 shows the PERIPHID6 Register bit assignments.

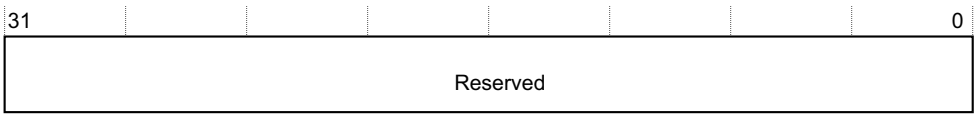


Figure 3-41 PERIPHID6 Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-42 shows the PERIPHID6 Register bit assignments.

Table 3-42 PERIPHID6 Register bit assignments

Bits	Name	Function
[31:0]	Reserved	Reserved

3.3.42 Peripheral ID7 Register

The PERIPHID7 Register characteristics are:

Purpose Reserved.

Configurations Present in all configurations.

Figure 3-42 shows the PERIPHID7 Register bit assignments.

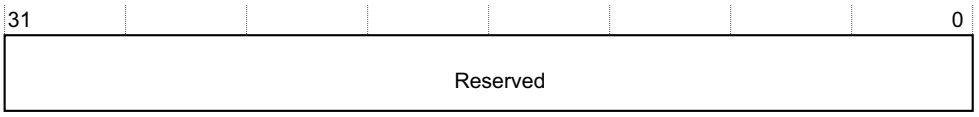


Figure 3-42 PERIPHID7 Register bit assignments

Table 3-43 shows the PERIPHID7 Register bit assignments.

Table 3-43 PERIPHID7 Register bit assignments

Bits	Name	Function
[31:0]	Reserved	Reserved

3.3.43 Peripheral ID0 Register

The PERIPHID0 Register characteristics are:

Purpose Part of the set of Peripheral Identification registers. Contains part of the designer-specific part number.

Configurations Present in all configurations.

Figure 3-43 shows the PERIPHID0 Register bit assignments.

Figure 3-43 PERIPHID0 Register bit assignments

Table 3-44 shows the PERIPHID0 Register bit assignments.

Table 3-44 PERIPHID0 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:0]	Part_Number_bits7to0	Bits [7:0] of the component's part number. This is selected by the designer of the component. `0x61` Lowest eight bits of the Part Number (`0x961`).

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.44 Peripheral ID1 Register

The PERIPHID1 Register characteristics are:

Purpose Part of the set of Peripheral Identification registers. Contains part of the designer specific part number and part of the designer identity.

Configurations Present in all configurations.

Figure 3-44 shows the PERIPHID1 Register bit assignments.

Figure 3-44 PERIPHID1 Register bit assignments

Table 3-45 shows the PERIPHID1 Register bit assignments.

Table 3-45 PERIPHID1 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:4]	JEP106_bits3to0	Bits [3:0] of the JEDEC identity code, indicating the designer of the component and the continuation code: `0xB` Lowest four bits of the JEP106 Identity Code.
[3:0]	Part_Number_bits11to8	Bits [11:8] of the components part number. This is selected by the designer of the component. `0x9` Upper four bits of the Part Number (`0x961`).

3.3.45 Peripheral ID2 Register

The PERIPHID2 Register characteristics are:

Purpose Part of the set of Peripheral Identification registers. Contains part of the designer identity and the product revision.

Configurations Present in all configurations.

Figure 3-45 shows the PERIPHID2 Register bit assignments.

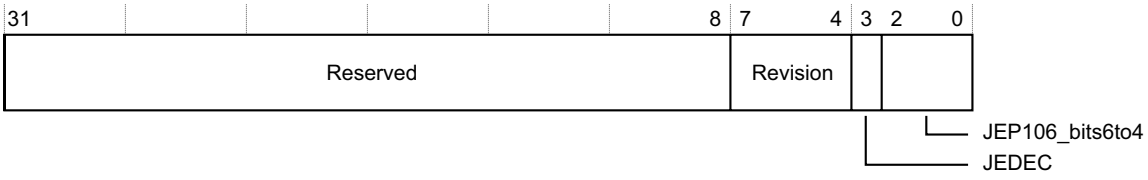


Figure 3-45 PERIPHID2 Register bit assignments

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-46 shows the PERIPHID2 Register bit assignments.

Table 3-46 PERIPHID2 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:4]	Revision	The Revision field is an incremental value starting at `0x0` for the first design of this component. This only increases by one for both major and minor revisions and is used as a look-up to establish the exact major or minor revision. `0x1` This device is at r0p1.
[3]	JEDEC	Always set. Indicates that a JEDEC assigned value is used. `0x1` The designer ID is specified by JEDEC.
[2:0]	JEP106_bits6to4	Bits [6:4] of the JEDEC-identity code, indicating the designer of the component and the continuation code. `0x3` Upper three bits of the JEP106 Identity Code.

3.3.46 Peripheral ID3 Register

The PERIPHID3 Register characteristics are:

Purpose Part of the set of Peripheral Identification registers. Contains the RevAnd and Customer Modified fields.

Configurations Present in all configurations.

Figure 3-46 shows the PERIPHID3 Register bit assignments.

Figure 3-46 PERIPHID3 Register bit assignments

Table 3-47 shows the PERIPHID3 Register bit assignments.

Table 3-47 PERIPHID3 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[3:0]	Customer_Modified	If the component is reusable IP, this value indicates whether the customer has modified the behavior of the component. In most cases, this field is zero. `0x0` Indicates that no modifications have been made.
[7:4]	RevAnd	This field indicates the minor errata fixes specific to this design. For example, metal fixes after implementation. In most cases, this field is zero. It is recommended that the component designer ensures that the metal fix changes this field, if required. For example, by driving it from registers that reset to zero. `0x0` Indicates that there have been no metal fixes to this component.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.47 Component ID0 Register

The COMPID0 Register characteristics are:

Purpose Indicates that the identification registers are present.

Configurations Present in all configurations.

Figure 3-47 shows the COMPID0 Register bit assignments.

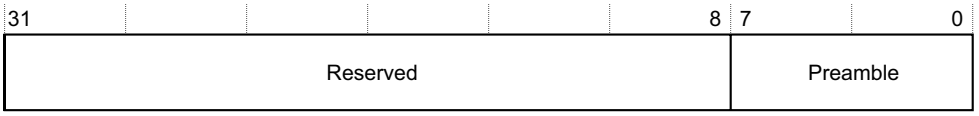


Figure 3-47 COMPID0 Register bit assignments

Table 3-48 shows the COMPID0 Register bit assignments.

Table 3-48 COMPID0 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:0]	Preamble	Contains bits [7:0] of the component identification. `0x0D` Identification value.

3.3.48 Component ID1 Register

The COMPID1 Register characteristics are:

Purpose Indicates that the identification registers are present, and also indicates the component class.

Configurations Present in all configurations.

Figure 3-48 shows the COMPID1 Register bit assignments.

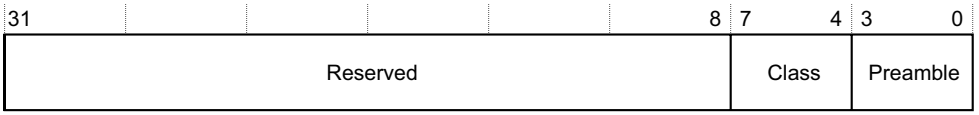


Figure 3-48 COMPID1 Register bit assignments

Table 3-49 shows the COMPID1 Register bit assignments.

Table 3-49 COMPID1 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[3:0]	Preamble	Contains bits [11:8] of the component identification. `0x0` Identification value.
[7:4]	Class	Class of the component. For example, ROM table, CoreSight component constitutes bits [15:12] of the component identification. `0x9` Indicates the component is a CoreSight component.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.3.49 Component ID2 Register

The COMPID2 Register characteristics are:

Purpose Indicates that the identification registers are present.

Configurations Present in all configurations.

Figure 3-49 shows the COMPID2 Register bit assignments.

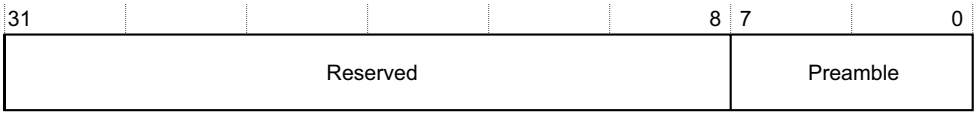


Figure 3-49 COMPID2 Register bit assignments

Table 3-50 shows the COMPID2 Register bit assignments.

Table 3-50 COMPID2 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:0]	Preamble	Contains bits [23:16] of the component identification. `0x05` Identification value.

3.3.50 Component ID3 Register

The COMPID3 Register characteristics are:

Purpose Indicates that the identification registers are present.

Configurations Present in all configurations.

Figure 3-50 shows the COMPID3 Register bit assignments.

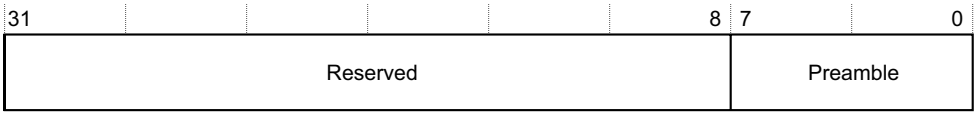


Figure 3-50 COMPID3 Register bit assignments

Table 3-51 shows the COMPID3 Register bit assignments.

Table 3-51 COMPID3 Register bit assignments

Bits	Name	Function
[31:8]	Reserved	Reserved.
[7:0]	Preamble	Contains bits [31:24] of the component identification. `0xB1` Identification value.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

3.4 Register access dependencies

The following sections describe when the TMC registers can be read or written:

- *Writes to TMC registers*
- *Reads from TMC registers* on page 3-51.

3.4.1 Writes to TMC registers

Table 3-52 shows the conditions necessary to write to each TMC register. Writing to the TMC under conditions other than those listed results in Unpredictable behavior. In Table 3-52, X indicates that any value is permitted.

Table 3-52 Permitted register writes

Write accesses to TMC registers				
Register	**TraceCaptEn**	**TMCReady**	**MODE**	**SG**
RSZ[a]	0	1	X	X
STS - MemErr	X	1	X	X
RRD	Read-only			
RRP	0	1	X	X
RWP	0	1	X	X
TRG	0	1	X	X
CTL	X	X	X	X
RWD	0	1	X	X
MODE	0	1	X	X
LBUFLEVEL	Read-only			
CBUFLEVEL	Read-only			
BUFWM	0	1	X	X
RRPHI	0	1	X	X
RWPHI	0	1	X	X
AXICTL	0	1	X	X
DBALO	0	1	X	X
DBAHI	0	1	X	X
FFSR	Read-only			
FFCR - DrainBuffer[b]	1	1	CB	-[c]
FFCR – StopOnTrigEvt	1	X	CB	X
	0	X	X	X
FFCR – StopOnFl	X	X	X	X
FFCR – TrigOnFl	X	X	X	X

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-52 Permitted register writes (continued)

Write accesses to TMC registers				
Register	**TraceCaptEn**	**TMCReady**	**MODE**	**SG**
FFCR – TrigOnTrigEvt	1	X	CB	X
	0	X	X	X
FFCR – TrigOnTrigIn	X	X	X	X
FFCR – FlushMan	X	X	X	X
FFCR – FOnTrigEvt	1	X	CB	X
	0	X	X	X
FFCR – FOnFlIn	X	X	X	X
FFCR – EnTI	0	1	X	X
FFCR – EnFt	0	1	X	X
PSCR	0	1	X	X
ITATBMDATA0	0	1	X	X
ITATBMCTR2	Read-only			
ITATBMCTR1	0	1	X	X
ITATBMCTR0	0	1	X	X
ITMISCOP0	0	1	X	X
ITTRFLIN	Read-only			
ITATBDATA0	Read-only			
ITATBCTR2	0	1	X	X
ITATBCTR1	Read-only			
ITATBCTR0	Read-only			
ITCTRL	0	1	X	X
CLAIMSET	X	X	X	X
CLAIMCLR	X	X	X	X
LOCKACCESS	X	X	X	X
LOCKSTATUS	Read-only			
AUTHSTATUS	Read-only			
DEVID	Read-only			
DEVTYPE	Read-only			
PERIPHID4	Read-only			
PERIPHID5	Read-only			
PERIPHID6	Read-only			

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-52 Permitted register writes (continued)

Write accesses to TMC registers				
Register	**TraceCaptEn**	**TMCReady**	**MODE**	**SG**
PERIPHID7	Read-only			
PERIPHID0	Read-only			
PERIPHID1	Read-only			
PERIPHID2	Read-only			
PERIPHID3	Read-only			
COMPID0	Read-only			
COMPID1	Read-only			
COMPID2	Read-only			
COMPID3	Read-only			

a. RO in ETR configuration, and RW in ETB and ETF configurations.
b. ETF configuration only.
c. SG applicable in ETR configuration only.

3.4.2 Reads from TMC registers

Table 3-53 lists the conditions under which read accesses from TMC registers return valid values. Reads at other times returns Unknown values. In the Table 3-53, X indicates that any value is permitted.

Table 3-53 Permitted register reads

Read accesses to TMC registers						
Register	**TraceCaptEn**	**TMCReady**	**MODE**	**SG**	**ITCTRL**	**Remarks**
RSZ[a]	X	X	X	X	X	-
STS-MemErr	X	X	X	X	0	-
STS-Empty	1	X	X	X	0	-
STS-FtEmpty	X	X	X	X	0	-
STS-TMCReady	X	X	X	X	0	-
STS-Triggered	1	X	CB	X	0	-
	0	X	X	X	0	Value of this bit when trace capture stops is held.
STS-Full	X	X	X	X	0	Value of this bit when trace capture stops is held.
RRD	0	1	X	X	0	-
	1	X	SWF	X	0	If trace memory is empty, data returned is `0xFFFFFFFF`.
	1	1	CB	X	0	

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-53 Permitted register reads (continued)

Read accesses to TMC registers						
Register	**TraceCaptEn**	**TMCReady**	**MODE**	**SG**	**ITCTRL**	**Remarks**
RRP	1	0	SWF	X	0	-
	1	1	SWF CB	X	0	-
	0	X	X	X	0	-
RWP	1	0	SWF	X	0	-
	1	1	SWF CB	X	0	-
	0	X	X	X	0	-
TRG	1	X	CB	X	0	The trigger counter is active only in Circular Buffer mode.
	0	X	X	X	0	
CTL	X	X	X	X	0	-
RWD	Write-only					
MODE	1	X	X	X	0	-
LBUFLEVEL	1	X	X	0	0	-
	0	X	X	X	0	Value of this register when trace capture stops is held.
CBUFLEVEL	1	X	X	0	0	-
BUFWM	X	X	X	X	X	Programmed registers can be read at any time. The return value is the value that was programmed.
RRPHI	1	0	SWF	X	0	-
	1	1	SWF CB	X	0	-
	0	X	X	X	0	-
RWPHI	1	0	SWF	X	0	-
	1	1	SWF CB	X	0	-
	0	X	X	X	0	-
AXICTL	X	X	X	X	X	-
DBALO	X	X	X	X	X	-
DBAHI	X	X	X	X	X	-
FFSR	X	X	X	X	0	-
FFCR	X	X	X	X	X	-
PSCR	X	X	X	X	X	-

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table 3-53 Permitted register reads (continued)

Read accesses to TMC registers						
Register	**TraceCaptEn**	**TMCReady**	**MODE**	**SG**	**ITCTRL**	**Remarks**
ITATBMDATA0	Write-only					
ITATBMCTR2	X	X	X	X	1	-
ITATBMCTR1	X	X	X	X	1	-
ITATBMCTR0	Write-only					
ITMISCOP0	Write-only					
ITTRFLIN	X	X	X	X	1	-
ITATBDATA0	X	X	X	X	1	-
ITATBCTR2	Write-only					
ITATBCTR1	X	X	X	X	1	-
ITATBCTR0	X	X	X	X	1	-
ITCTRL	X	X	X	X	X	-
CLAIMSET	X	X	X	X	X	-
CLAIMCLR	X	X	X	X	X	-
LOCKACCESS	Write-only					
LOCKSTATUS	X	X	X	X	X	-
AUTHSTATUS	X	X	X	X	X	-
DEVID	X	X	X	X	X	-
DEVTYPE	X	X	X	X	X	-
PERIPHID4	X	X	X	X	X	-
PERIPHID5	X	X	X	X	X	-
PERIPHID6	X	X	X	X	X	-
PERIPHID7	X	X	X	X	X	-
PERIPHID0	X	X	X	X	X	-
PERIPHID1	X	X	X	X	X	-
PERIPHID2	X	X	X	X	X	-
PERIPHID3	X	X	X	X	X	-
COMPID0	X	X	X	X	X	-
COMPID1	X	X	X	X	X	-
COMPID2	X	X	X	X	X	-
COMPID3	X	X	X	X	X	-

a. In ETR configuration.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Appendix A
Signal Descriptions

This appendix describes the signals used in the TMC. It contains the following sections:

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.1 Clocks and resets

Table A-1 shows the clock and reset signals.

Table A-1 Clock and reset signals

Signal	Type	Description
CLK	Input	TMC clock. Clocks all the interfaces in the TMC.
RESETn	Input	TMC reset. Resets all the interfaces and internal registers in the TMC.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.2 ATB interface signals

The following section describes the ATB interface signals:

- *ATB master interface signals*
- *ATB slave interface signals*.

For more information, see the *AMBA 3 ATB Protocol Specification*.

A.2.1 ATB master interface signals

Trace output stream. This interface is present only in ETF configurations. Table A-2 shows the ATB master interface signals.

Table A-2 ATB master interface signals

Signal	Type	Description
ATVALIDM	Output	Valid signals in this cycle from the trace source
ATREADYM	Input	If there is valid data, that is, **ATVALID** HIGH, the data was accepted this cycle
ATIDM[6:0]	Output	Trace source ID
ATBYTESM[a]	Output	Number of valid bytes on **ATDATA**, minus one
ATDATAM[b]	Output	Trace data, LSB aligned
AFVALIDM	Input	Any data remaining in any buffers must be flushed
AFREADYM	Output	Data flush complete, **AFVALID** can be deasserted

a. The width is dependent on the configured ATB width:
[1:0] 32-bit ATB.
[2:0] 64-bit ATB.
[3:0] 128-bit ATB.
b. The width is dependent on the configured ATB width:
[31:0] 32-bit ATB
[63:0] 64-bit ATB
[127:0] 128-bit ATB

A.2.2 ATB slave interface signals

Trace input stream. Table A-3 shows the ATB slave interface signals.

Table A-3 ATB slave interface signals

Signal	Type	Description
ATVALIDS	Input	Valid signals in this cycle from the trace source
ATREADYS	Output	If there is valid data, that is, **ATVALID** HIGH, the data was accepted this cycle
ATIDS[6:0]	Input	Trace source ID
ATBYTESS[a]	Input	Number of valid bytes on **ATDATA**, minus one
ATDATAS[b]	Input	Trace data, LSB aligned
AFVALIDS	Output	Any data remaining in any buffers must be flushed
AFREADYS	Input	Data flush complete, **AFVALID** can be deasserted

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

a. The width is dependent on the configured ATB width:
[1:0] 32-bit ATB.
[2:0] 64-bit ATB.
[3:0] 128-bit ATB.
b. The width is dependent on the configured ATB width:
[31:0] 32-bit ATB
[63:0] 64-bit ATB
[127:0] 128-bit ATB

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.3 APB signals

The programming interface for the TMC control. Table A-4 shows the APB signals. For more information, see *AMBA 3 APB Protocol Specification*.

Table A-4 APB signals

Signal	Type	Description
PCLKENDBG	Input	Clock enable for APB.
PADDRDBG[11:2]	Input	Programming address. Occupies a 4KB region of memory.
PADDRDBG31	Input	HIGH for external accesses to bypass the Lock Access mechanism. Wire to **PADDRDBG[31]** in systems.
PSELDBG	Input	Indicates this device is currently being accessed.
PENABLEDBG	Input	Indicates second, and subsequent, cycles of a APB transfer.
PWRITEDBG	Input	This access is a write transfer.
PRDATADBG[31:0]	Output	Read data bus.
PWDATADBG[31:0]	Input	Write data bus.
PREADYDBG	Output	The ready signal used by the slave to extend an APB transfer.
PSLVERRDBG	Output	Error response of the APB interface.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.4 SRAM signals

The SRAM Interface is for the memory access. This interface is only present in ETB and ETF configurations. Table A-5 shows the SRAM signals.

Table A-5 SRAM signals

Signal	Type	Description
MEMCEN	Output	Chip enable, active LOW.
MEMADDR[a]	Output	Address.
MEMWEN	Output	Write enable, active LOW.
MEMD[b]	Input	Write data. The data is valid in the same cycle as the address.
MEMQ[b]	Output	Read data. The data is returned one cycle after the address.

a. The width of this bus depends on the memory size and width.
b. The width is dependent on the configured ATB width:
[63:0] 32-bit ATB
[127:0] 64-bit ATB
[255:0] 128-bit ATB

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.5 AXI signals

This interface is present only in ETR configurations. For more information, see the *AMBA 3 AXI Protocol Specification*. Table A-6 shows the AXI signals.

Table A-6 AXI signals

Signal	Type	Source	Description
AWADDRM[39:0]	Output	Master	Write address.
AWLENM[3:0]	Output	Master	Burst length.
AWSIZEM[2:0]	Output	Master	Burst size.
AWBURSTM[1:0]	Output	Master	Burst type.
AWLOCKM[1:0]	Output	Master	Lock type.
AWCACHEM[3:0]	Output	Master	Cache type.
AWPROTM[2:0]	Output	Master	Protection type.
AWVALIDM	Output	Master	Write address valid.
AWREADYM	Input	Slave	Write address ready.
WDATAM[a]	Output	Master	Write data.
WSTRBM[b]	Output	Master	Write strobes.
WLASTM	Output	Master	Write last.
WVALIDM	Output	Master	Write valid.
WREADYM	Input	Slave	Write ready.
BRESPM[1:0]	Input	Slave	Write response.
BVALIDM	Input	Slave	Write response valid.
BREADYM	Output	Master	Response ready.
ARADDRM[39:0]	Output	Master	Read address.
ARLENM[3:0]	Output	Master	Burst length.
ARSIZEM[2:0]	Output	Master	Burst size.
ARBURSTM[1:0]	Output	Master	Burst type.
ARLOCKM[1:0]	Output	Master	Lock type.
ARCACHEM[3:0]	Output	Master	Cache type.
ARPROTM[2:0]	Output	Master	Protection type.
ARVALIDM	Output	Master	Read address valid.
ARREADYM	Input	Slave	Read address ready.
RDATAM[a]	Input	Slave	Read data.
RRESPM[1:0]	Input	Slave	Read response.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Table A-6 AXI signals (continued)

Signal	Type	Source	Description
RLASTM	Input	Slave	Read last.
RVALIDM	Input	Slave	Read valid.
RREADYM	Output	Master	Read ready.

a. The width is dependent on the configured AXI width:
[31:0] 32-bit AXI
[63:0] 64-bit AXI
[127:0] 128-bit AXI
b. The width is dependent on the configured AXI width:
[3:0] 32-bit AXI
[7:0] 64-bit AXI
[15:0] 128-bit AXI

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.6 Authentication signals

The Authentication interface is for disabling AXI accesses. This interface is present only in ETR configurations. For more information, see the *CoreSight Architecture Specification*.

Table A-7 Authentication signals

Signal	Type	Description
DBGEN	Input	Permits transfers to take place on the AXI master interface
SPIDEN	Input	Permits secure transfers to take place on the AXI master interface

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.7 Cross-trigger interface

This interface contains signals for connecting to a CTI.

Table A-8 Cross-trigger interface signals

Signal	Type	Description
TRIGIN	Input	This input can cause a Trigger Event
FLUSHIN	Input	This input can cause a trace flush
FULL	Output	When the TMC is not in integration mode, this output indicates the value of the Full bit in the STS Register
ACQCOMP	Output	When the TMC is not in integration mode, this output indicates the value of the FtEmpty bit in the STS Register

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.8 Synchronization request interface

This interface controls the frequency of synchronization information.

Table A-9 Synchronization request interface signals

Signal	Type	Description
SYNCREQM[a]	Input	Synchronization request on ATB master interface
SYNCREQS	Output	Synchronization request on ATB slave interface

a. **SYNCREQM** is present only in ETF configuration.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.9 Low-power interface signals

This interface is present only in ETR configurations. Table A-10 shows the low-power interface signals.

Table A-10 Low-power interface signals

Signal	Type	Description
CSYSREQ	Input	Requests an AXI master interface stall, active LOW.
CSYSACK	Output	Indicates that the AXI master interface is stalled, active LOW.
CACTIVE	Output	Always LOW. Indicates that a low-power request is accepted.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

A.10 Test interface signals

Table A-11 shows the test interface signals.

Table A-11 Test interface signals

Signal	Type	Description
DFTTESTMODE	Input	Test mode. This signal is used to force the STM clock on in test mode.
DFTCLKDISABLE	Input	Test clock disable. This signal is used to switch off the STM clock in the capture phase of DFT testing of other components in the system.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Appendix B
Revisions

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

Table B-1 Issue A

Change	Location	Affects
No changes, first release	-	-

Table B-2 Difference between Issue A and Issue B

Change	Location	Affects
Updated PERIPHID2 reset value	Table 3-1 on page 3-3 and Table 3-46 on page 3-46	r0p1
Updated revision value	Table 3-46 on page 3-46	r0p1

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Glossary

This glossary describes some of the terms used in technical documents from ARM.

Advanced eXtensible Interface (AXI)
A bus protocol that supports separate phases for address or control and data, unaligned data transfers using byte strobes, burst-based transactions with only start address issued, separate read and write data channels, issuing multiple outstanding addresses, out-of-order transaction completion, and easy addition of register stages to provide timing closure.

The AXI protocol includes optional extensions for signaling for low-power operation.

Advanced Microcontroller Bus Architecture (AMBA)
The AMBA family of protocol specifications is the ARM open standard for on-chip buses. AMBA provides a strategy for the interconnection and management of the functional blocks that make up a *System-on-Chip* (SoC). Applications include the development of embedded systems with one or more processors or signal processors and multiple peripherals. AMBA defines a common backbone for SoC modules, and therefore complements a reusable design methodology.

Advanced Peripheral Bus (APB)
A bus protocol that is designed for use with ancillary or general-purpose peripherals such as timers, interrupt controllers, UARTs, and I/O ports. It connects to the main system bus through a system-to-peripheral bus bridge that helps reduce system power consumption.

Advanced Trace Bus (ATB)
A bus used by trace devices to share CoreSight capture resources.

Aligned A data item stored at an address that is divisible by the number of bytes that defines its data size is said to be aligned. Aligned doublewords, words, and halfwords have addresses that are divisible by eight, four, and two respectively. The terms doubleword-aligned, word-aligned, and

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

halfword-aligned therefore stipulate addresses that are divisible by eight, four, and two respectively. An aligned access is one where the address of the access is aligned to the size of an element of the access.

AMBA *See* Advanced Microcontroller Bus Architecture.

APB *See* Advanced Peripheral Bus.

ATB *See* Advanced Trace Bus.

AXI *See* Advanced eXtensible Interface.

AXI channels, channel order and interfaces

The block diagram shows:

- the order in which AXI channel signals are described
- the master and slave interface conventions for AXI components.

AXI signal names have a one or two letter prefix that denotes the AXI channel as follows:

AW Write address channel.

W Write data channel.

B Write response channel.

AR Read address channel.

R Read data channel.

General descriptions of AXI signals use x to represent this prefix, for example, **xVALID** and **xREADY**.

Beat Alternative word for an individual transfer within a burst. For example, an INCR4 burst comprises four beats.

See also Burst.

Burst A group of transfers to consecutive addresses. Because the addresses are consecutive, the device transmitting the data does not have to supply an address for any transfer after the first one. This increases the speed at which the burst occurs. If using an AMBA interface, the transmitting device controls the burst using signals that indicate the length of the burst and how the addresses are incremented.

See also Beat.

CoreSight ARM on-chip debug and trace components, that provide the infrastructure for monitoring, tracing, and debugging a complete system on chip.

See also CoreSight ECT, CoreSight ETB, CoreSight ETM, Trace Funnel, and *Trace Port Interface Unit* (TPIU).

CoreSight ETB CoreSight ETB is a trace sink that provides on-chip storage of trace data using a configurable sized RAM.

See also CoreSight, CoreSight ETB, Embedded Trace Buffer, and Embedded Trace Macrocell.

Cross Trigger Interface (CTI)

Part of an *Embedded Cross Trigger* (ECT) device. In an ECT, the CTI provides the interface between a processor or ETM and the CTM.

CTI *See* Cross Trigger Interface.

DAP *See* Debug Access Port.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Debug Access Port (DAP)
A block that acts as a master on a system bus and provides access to the bus from an external debugger.

Debugger A debugging system that includes a program, used to detect, locate, and correct software faults, together with custom hardware that supports software debugging.

Embedded Trace Buffer (ETB)
Provides on-chip storage of trace data using a configurable sized RAM.

Embedded Trace Macrocell (ETM)
A hardware macrocell that, when connected to a processor, outputs trace information on a trace port. The ETM provides processor driven trace through a trace port compliant to the ATB protocol. An ETM always supports instruction trace, and might support data trace.

ETB *See* Embedded Trace Buffer.

ETM *See* Embedded Trace Macrocell.

Formatter In an ETB or TPIU, an internal input block that embeds the trace source ID in the data to create a single trace stream.

Host A computer that provides data and other services to another computer. Especially, a computer providing debugging services to a target being debugged.

JTAG Access Port (JTAG-AP)
An optional component of the DAP that provides debugger access to on-chip scan chains.

Macrocell A complex logic block with a defined interface and behavior. A typical VLSI system comprises several macrocells, such as a processor, an ETM, and a memory block integrated with application-specific logic.

PA *See* Physical Address.

Physical Address (PA)
The address that identifies a main memory location.

Prefetch abort An indication from a memory system to the processor that an instruction has been fetched from an illegal memory location. An exception must be taken if the processor attempts to execute the instruction. A Prefetch abort can be caused by the external or internal memory system as a result of attempting to access invalid instruction memory.

See also Data abort, External abort and Abort.

Read Memory operations that have the semantics of a load. See the *ARM Architecture Reference Manual* for more information.

Replicator In an ARM trace macrocell, a replicator enables two trace sinks to be wired together and to operate independently on the same incoming trace stream. The input trace stream is output onto two independent ATB ports.

Reserved Registers and instructions that are reserved are Unpredictable unless otherwise stated. Bit positions described as Reserved are UNK/SBZP.

SBZ *See* Should Be Zero.

SBZP *See* Should Be Zero or Preserved.

Serial Wire Debug (SWD)
A debug implementation that uses a serial connection between the SoC and a debugger.

This connection normally requires a bi-directional data signal and a separate clock signal, rather than the four to six signals required for a JTAG connection.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential

Serial Wire Debug Port (SWDP)
The interface for Serial Wire Debug.

Should Be Zero (SBZ)
Software must write as 0, or all 0s for bit fields. Writing any other value produces Unpredictable results.

Should Be Zero or Preserved (SBZP)
Software must write as 0, or all 0s for a bit field, if the value is being written without having previously been read, or if the register has not been initialized. If the register has previously been read, software must preserve the field value by writing back the value that was read from the same field on the same processor.

SWD *See* Serial Wire Debug.

SWDP *See* Serial Wire Debug Port.

TPA *See* Trace Port Analyzer.

TPIU *See* Trace Port Interface Unit.

Trace funnel In CoreSight, a device that combines multiple trace sources onto a single bus.

See also CoreSight and Embedded Trace Macrocell.

Trace port A port on a device, such as a processor or ASIC, used to output trace information.

Trace Port Analyzer (TPA)
A hardware device that captures trace information output on a trace port. This can be a low-cost product designed specifically for trace acquisition, or a logic analyzer.

Trace Port Interface Unit (TPIU)
Drains trace data and acts as a bridge between the on-chip trace data and the data stream captured by a TPA.

Unknown An Unknown value does not contain valid data, and can vary from moment to moment, instruction to instruction, and implementation to implementation. An Unknown value must not be a security hole.

UNP *See* Unpredictable.

Unpredictable For a processor means the behavior cannot be relied on. Unpredictable behavior must not represent security holes. Unpredictable behavior must not halt or hang the processor, or any parts of the system.

Word A 32-bit data item. Words are normally word-aligned in ARM systems.

Write Operations that have the semantics of a store. See the *ARM Architecture Reference Manual* for more information.

Write buffer A block of high-speed memory implemented to optimize stores to main memory.

Write interleave capability
The number of data-active write transactions for which the interface can transmit data. This is counted from the earliest transaction.

Write interleave depth
The number of data-active write transactions for which the interface can receive data.

Copyright © 2010 ARM. All rights reserved.
Non-Confidential