CoreSight™ ETM™-R4

Revision: r2p1

Technical Reference Manual

ARM DDI 0367D (ID041111)

CoreSight ETM-R4

Technical Reference Manual

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
18 January 2006	A	Confidential	First release, for r0p0
23 November 2007	B	Non-Confidential	First release for r1p0
26 March 2009	C	Non-Confidential	First release for r2p0
06 April 2011	D	Non-Confidential	First release for r2p1

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 ETM-R4 Technical Reference Manual

Chapter 3 Programmers Model

Appendix A Signal Descriptions

Appendix B AC Characteristics

Appendix C Revisions

Preface

This preface introduces the *CoreSight ETM-R4 Technical Reference Manual*. It contains the following sections:

- *About this book* on page vi
- *Feedback* on page x.

About this book

This book is for the CoreSight Embedded Trace Macrocell for the Cortex™-R4 and Cortex-R4F processors.

You implement the ETM-R4 macrocell with the Cortex-R4 processor or the Cortex-R4F processor. In this manual, in general:

- Reference to the processor applies to the Cortex-R4 processor or the Cortex-R4F processor.
- Reference to the Cortex-R4 processor applies also to the Cortex-R4F processor.

The context makes it clear if information applies to only one of the processor options.

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:

- Designers of development tools providing support for ETM functionality. Implementation-specific behavior is described in this document. You can find complementary information in the *Embedded Trace Macrocell Architecture Specification*.
- Hardware and software engineers integrating the macrocell into an ASIC that includes a Cortex-R4 processor. You can find complementary information in the *CoreSight ETM-R4 Integration Manual*.

Using this book

This book is organized into the following chapters:

Chapter 1 *Introduction*

Read this for an introduction to the functionality of the macrocell.

Chapter 2 *Functional Description*

Read this for a description of the interfaces, operation, clocking and resets of the macrocell.

Chapter 3 *Programmers Model*

Read this for a description of the programmers model for the macrocell.

Appendix A *Signal Descriptions*

Read this for a description of the signals used in the macrocell.

Appendix B *AC Characteristics*

Read this for a description of the macrocell input and output signal timing.

Appendix C *Revisions*

Read this for a description of the technical changes between released issues of this book.

Conventions

Conventions that this book can use are described in:

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

Typographical

The typographical conventions are:

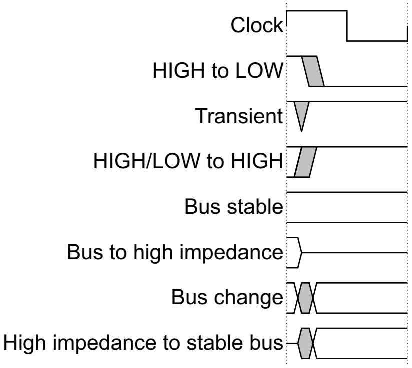
italic	Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at 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

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.

Prefix A Denotes global *Advanced eXtensible Interface* (AXI) signals.

Prefix AF Denotes *Advanced Trace Bus* (ATB) flush control signals.

Prefix AR Denotes AXI read address channel signals.

Prefix AT	Denotes ATB data flow signals.
Prefix AW	Denotes AXI write address channel signals.
Prefix B	Denotes AXI write response channel signals.
Prefix C	Denotes AXI low-power interface signals.
Prefix H	Denotes *Advanced High-performance Bus* (AHB) signals.
Prefix P	Denotes *Advanced Peripheral Bus* (APB) signals.
Prefix R	Denotes AXI read data channel signals.
Prefix W	Denotes AXI write data channel signals.

Additional reading

This section lists publications by ARM and by third parties.

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

See the glossary, http://infocenter.arm.com/help/topic/com.arm.doc.aeg0014-/index.html, for a list of terms and acronyms specific to ARM.

See onARM, `http://onarm.com`, for embedded software development resources including the *Cortex Microcontroller Software Interface Standard* (CMSIS).

ARM publications

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

- *CoreSight ETM-R4 Configuration and Sign-off Guide (*ARM DII 0132)
- *CoreSight ETM-R4 Integration Manual* (ARM DII 0133)
- *Embedded Trace Macrocell™ Architecture Specification* (ARM IHI 0014)
- *CoreSight Components Technical Reference Manual* (ARM DDI 0314)
- *CoreSight Design Kit R4 Integration Manual* (ARM DII 0134)
- *ARM Reference Peripheral Specification* (ARM DDI 0062)
- *Cortex-R4 Technical Reference Manual* (ARM DDI 0363)
- *CoreSight Architecture Specification* (ARM IHI 0020)
- *CoreSight Technology System Design Guide* (ARM DGI 0012)
- *AMBA™ 3 APB Protocol Specification* (ARM IHI 0024)
- *AMBA 3 ATB Protocol Specification* (ARM IHI 0032).

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 0367D
- the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Chapter 1
Introduction

This chapter introduces the ETM-R4 macrocell. It contains the following sections:

1.1 About the ETM-R4

The ETM-R4 macrocell provides real-time instruction trace and data trace for the Cortex-R4 microprocessor. The ETM-R4 generates information that trace software tools use to reconstruct the execution of all or part of a program.

For full reconstruction of program execution, the ETM-R4 is able to trace:

- all instructions, including condition code pass/fail and dual issue information
- load/store address and data values
- data values used in coprocessor register transfers
- values of context-ID changes
- target addresses of taken direct and indirect branch operations
- exceptions
- changes in processor instruction set state
- entry to and return from Debug state when Halting Debug-mode is enabled
- cycle counts between executed instructions.

The ETM-R4 contains logic, known as resources, that enables you to control tracing by specifying the exact set of triggering and filtering conditions required for a particular application. Resources include address comparators and data value comparators, counters, and sequencers.

The ETM-R4 is a CoreSight component, and is an integral part of the ARM Real-time Debug solution, RealView®. For more information about CoreSight, see the *CoreSight Architecture Specification* and *CoreSight Technology System Design Guide*. For more information about the ETM architecture, see the *Embedded Trace Macrocell Architecture Specification*.

1.1.1 The CoreSight debug environment

The ETM-R4 is designed for use with CoreSight, an extensible, system-wide debug and trace architecture from ARM. See the *CoreSight Design Kit R4 Integration Manual* for more information about how to use the ETM-R4 in a full CoreSight system. See the *CoreSight ETM-R4 Integration Manual* for an example of how to use the ETM-R4 in a simple trace system.

A software debugger provides the user interface to the ETM-R4. You can use this interface to:

- configure ETM-R4 facilities such as filtering
- configure optional trace features such as cycle accurate tracing

- configure the other CoreSight components such as the *Trace Port Interface Unit* (TPIU)
- access the processor debug and performance monitor units.

A CoreSight system can provide memory mapped access from the processor to its own debug and trace components.

The ETM-R4 outputs its trace stream to the AMBA 3 ATB interface. The CoreSight infrastructure provides the following options:

- Export the trace information through a trace port. An external *Trace Port Analyzer* (TPA) captures the trace information as Figure 1-1 on page 1-4 shows.
- Write the trace information directly to an on-chip *Embedded Trace Buffer* (ETB). You can read out the trace at low speed using a JTAG or Serial Wire interface when the trace capture is complete as Figure 1-1 on page 1-4 shows.

The debugger extracts the captured trace information from the TPA or ETB and decompresses it to provide full disassembly, with symbols, of the code that was executed. The trace information generated by the ETM-R4 gives the debugger the capability to link this data back to the original high-level source code, to provide a visualization of how the code was executed on the Cortex-R4 processor.

Figure 1-1 on page 1-4 shows how the ETM-R4 fits into a CoreSight debug environment to provide full trace capabilities in a single processor system. The external debug software configures the trace and debug components through the DAP. The ROM table contains a unique identification code for the SoC and the base addresses of the components connected to the debug APB. The trace stream from the ETM-R4 is replicated to provide on-chip storage using the CoreSight ETB or output off-chip using the TPIU. Cross-triggering operates through the CTIs and the cross-trigger matrix.

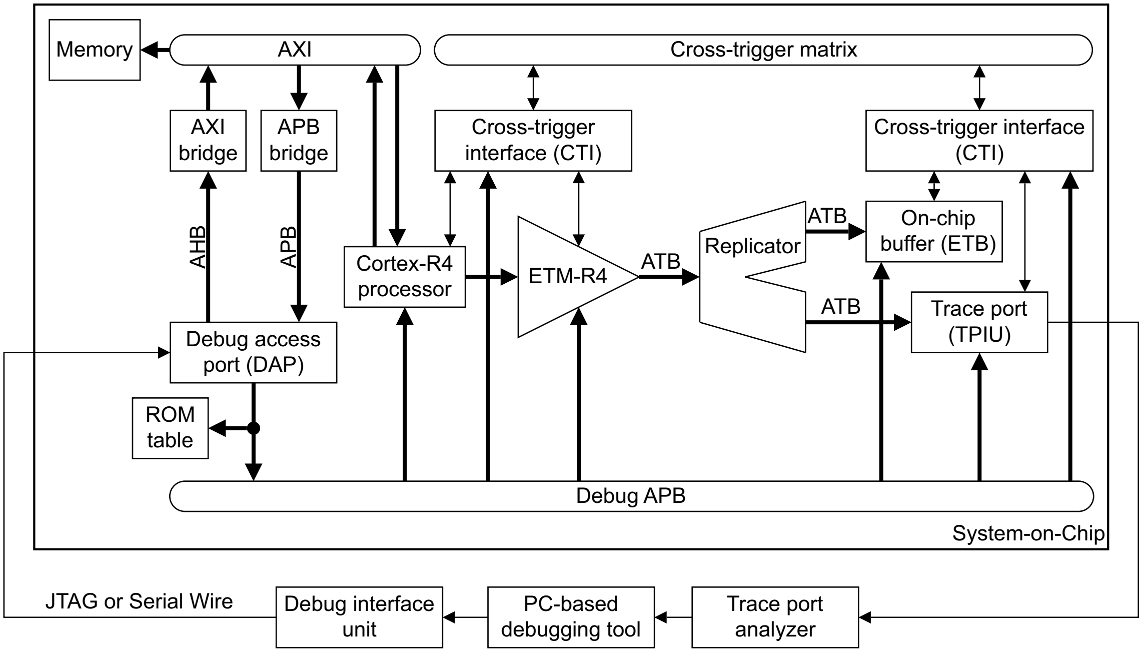


Figure 1-1 ETM-R4 system diagram

Note

In Figure 1-1, the arrows on the thick lines show the transaction direction on busses, from master to slave port. Each bus contains individual signals that go from master to slave and other signals that go from slave to master.

1.2 Compliance

The ETM-R4 Embedded Trace Macrocell complies with, or implements, the specifications described in:

- *Trace macrocell*
- *Advanced Microcontroller Bus Architecture*.

This TRM complements architecture reference manuals, architecture specifications, protocol specifications, and relevant external standards. It does not duplicate information from these sources.

1.2.1 Trace macrocell

The ETM-R4 Embedded Trace Macrocell implements the ETM v3.3 architecture profile. See the *CoreSight ETM-R4 Technical Reference Manual*.

1.2.2 Advanced Microcontroller Bus Architecture

This ETM-R4 Embedded Trace Macrocell complies with the AMBA 3 protocol. See *AMBA AXI Protocol Specification* and *AMBA 3 APB Protocol Specification*.

1.3 Features

ETM-R4 supports tracing of 32-bit ARM instructions, and 16-bit and 32-bit Thumb instructions.

See the *Embedded Trace Macrocell Architecture Specification* for information about:

- the trace protocol
- the features of ETMv3.3
- controlling tracing using triggering and filtering resources
- ETM sharing.

Table 1-1 lists the features of the ETM-R4 that are implementation-defined, in terms of either:

- the number of times the feature is implemented
- the size of the feature.

Table 1-1 ETM-R4 features with implementation-defined number of instances or size

Feature	ETM-R4 value	Notes
Address comparators	4 pairs	See bits[3:0] of the ETMCCR[a]
Data value comparators	2	See bits[7:4] of the ETMCCR[a]
EmbeddedICE watchpoint comparators	Not implemented	Not supported in ETMv3.3
Context ID comparators	1	See bits[25:24] of the ETMCCR[a]
Counters	2	See bits[15:13] of the ETMCCR[a]
Sequencer	1	See bit[16] of the ETMCCR.[a]
Memory Map decoder inputs	Not implemented	See bits[12:8] of the ETMCCR[a]
External inputs	0-4	See bits[19:17] of the ETMCCR[a]
External outputs	0-2	See bits[22:20] of the ETMCCR[a]
Extended external input bus width	47	See bits[10:3] of the ETMCCER[b]
Extended external input selectors	2	See bits[2:0] of the ETMCCER[b]
Instrumentation resources	Not implemented	Not supported in ETMv3.3

Table 1-1 ETM-R4 features with implementation-defined number of instances or size (continued)

Feature	ETM-R4 value	Notes
Trace port size	32-bit	See bits[21,6:4] of the ETMCR[c]
FIFO size	144 bytes	-
ASICCTL general-purpose bus interface	8-bit	See ETMASICCR[d]

a. See *Configuration Code Register* on page 3-23.
b. See *Configuration Code Extension Register* on page 3-28.
c. See *ETM Main Control Register* on page 3-18.
d. See *ASIC Control Register* on page 3-26

Table 1-2 shows the optional features of the ETM architecture the ETM-R4 implements.

Table 1-2 ETM-R4 implementation of optional features

Feature	Implemented?	Notes
FIFOFULL control	No	See bit[23] of the ETMCCR[a]
Trace Start/Stop block	Yes	See bit[26] of the ETMCCR[a]
Trace all branches	Yes	See bit[8] of the ETMCR[b]
Cycle-accurate trace	Yes	See bit[12] of the ETMCR[b]
Data trace options		
Data address tracing	Yes	See bits[3:2] of the ETMCR[b]
Data value tracing	Yes	See bits[3:2] of the ETMCR[b]
Data-only tracing	Yes	See bit[20] of the ETMCR[b]
CPRT tracing	Yes	See bits[19, 1] of the ETMCR[b]
Data address comparison	Yes	Bit[12] of the ETMCCER[c] reads-as-zero
EmbeddedICE behavior control	No	Not supported in ETMv3.3
EmbeddedICE inputs to Trace Start/Stop block	No	Not supported in ETMv3.3
Alternative address compression	No	Not supported in ETMv3.3
OS Lock mechanism	No	Not implemented

Table 1-2 ETM-R4 implementation of optional features (continued)

Feature	Implemented?	Notes
Secure non-invasive debug	No	Cortex-R4 does not implement the Security Extensions
Context ID tracing	Yes	See bits[15:14] of the ETMCR[b]
Trace output	Yes	ATB

a. See *Configuration Code Register* on page 3-23.
b. See *ETM Main Control Register* on page 3-18.
c. See *Configuration Code Extension Register* on page 3-28.

See Appendix A *Signal Descriptions* for information about the macrocell signals.

1.4 Interfaces

The ETM-R4 has the following main interfaces:

- Processor trace
- ATB
- APB Debug
- Test.

Interfaces on page 2-4 describes the ETM-R4 interfaces in more detail.

1.5 Configurable options

The ETM-R4 macrocell includes the following configuration inputs:

- **MAXEXTOUT[1:0]** determines the maximum number of external outputs.
- **MAXEXTIN[2:0]** determines the maximum number of external inputs.
- **MAXCORES[2:0]** determines the number of processors that share the ETM.

You can read the MAXEXTOUT and MAXEXTIN values from bits[22:17] of the ETMCCR, see *Configuration Code Register* on page 3-23. You can read the MAXCORES value from bits[14:12] of the ETMSCR. see *System Configuration Register* in the *Embedded Trace Macrocell Architecture Specification*.

1.6 Test features

The ETM-R4 provides the **SE** and **RSTBYPASS** inputs for testing the implemented device. See the *CoreSight ETM-R4 Integration Manual*.

See also *Integration Test Registers* on page 3-37 for information about the integration test registers, provided for testing the ETM-R4 implementation in a SoC.

1.7 Product documentation, architecture and design flow

This section describes the ETM-R4 books, how they relate to the design flow, and the relevant architectural standards and protocols. It contains the following sections:

- *Documentation*
- *Design flow* on page 1-13.

See *Additional reading* on page ix for more information about the books described in this section.

1.7.1 Documentation

The ETM-R4 documentation is as follows:

Technical Reference Manual

The *Technical Reference Manual* (TRM) describes the functionality and the effects of functional options on the behavior of the ETM-R4. It is required at all stages of the design flow. The choices made in the design flow can mean that some behavior described in the TRM is not relevant. If you are programming the ETM-R4 then contact:

- the implementer to determine:
  - the build configuration of the implementation
  - what integration, if any, was performed before implementing the ETM-R4
- the integrator to determine the pin configuration of the device that you are using.

Configuration and Sign-off Guide

The *Configuration and Sign-off Guide* (CSG) describes:

- the available build configuration options and related issues in selecting them
- how to configure the *Register Transfer Level* (RTL) with the build configuration options
- how to integrate RAM arrays
- how to run test vectors
- the processes to sign off the configured design.

The ARM product deliverables include reference scripts and information about using them to implement your design. Reference methodology flows supplied by ARM are example reference implementations. Contact your EDA vendor for EDA tool support.

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

Integration Manual

The *Integration Manual* (IM) describes how to integrate the ETM-R4 into a SoC. It includes describing the pins that the integrator must tie off to configure the macrocell for the required integration. Some of the integration is affected by the configuration options used when implementing the ETM-R4.

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

1.7.2 Design flow

The ETM-R4 is delivered as synthesizable RTL. Before it can be used in a product, it must go through the following processes:

Implementation

The implementer configures and synthesizes the RTL to produce a hard macrocell. This might include integrating RAMs into the design.

Integration The integrator connects the implemented design into a SoC. This includes connecting it to a memory system and peripherals.

Programming

This is the last process. The system programmer develops the software required to configure and initialize the ETM-R4, and tests the required application software.

Each process:

- can be performed by a different party
- can include implementation and integration choices affect the behavior and features of the ETM-R4.

The operation of the final device depends on:

Build configuration

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

Configuration inputs

The integrator configures some features of the ETM-R4 by tying inputs to specific values. These configurations affect the start-up behavior before any software configuration is made. They can also limit the options available to the software.

Software configuration

The programmer configures the ETM-R4 by programming particular values into registers. This affects the behavior of the ETM-R4.

Note

This manual refers to implementation-defined features that are applicable to build configuration options. Reference to a feature that is included means that the appropriate build and pin configuration options are selected. Reference to an enabled feature means one that has also been configured by software.

1.8 Product revisions

You can integrate any revision of ETM-R4 with your chosen revision of Cortex-R4 or Cortex-R4F. However, ARM recommends that you use the most recent ETM-R4 revision for ease of integration and functional operation. This section describes the differences in functionality between product revisions:

r0p0-r1p0 Functional changes are:

- The **EVNTBUS** input signal width increases from 29 to 47 bits, see *Interaction with the Performance Monitoring Unit* on page 2-10.
- The number of valid bits in each Selection value field in the ETMEXTINSELR increases from 5 to 6, see *Extended External Input Selection Register* on page 3-30.
- An extra bit is defined in the ITETMIF Integration Test Register, to return the value of the **EVNTBUS[46]** signal, see *Processor-ETM Interface Register* on page 3-40.
- The value of the size of extended external input bus field in the ETMCCER register changes from 29 to 47, see *Configuration Code Extension Register* on page 3-28.
- File, module and macro names change to make them unique, to avoid clashes with the names used on other ETM.
- The revision fields in the ID registers change, to indicate the r1p0 revision:
  - The Implementation revision field of the ETMIDR changes to b0001, see *ETM ID Register* on page 3-27.
  - The Revision field of the ETMPIDR2 Register, changes to b0001, see *Peripheral Identification Registers* on page 3-31.

r1p0-r2p0 Functional changes are:

- The FIFO size changes from 72 bytes to 144 bytes. See Table 1-1 on page 1-6.
- The revision fields in the ID registers change, to indicate the r2p0 revision:
  - The implementation revision field of the ETMIDR register changes to b0010, see *ETM ID Register* on page 3-27.
  - The Revision field of the ETMPIDR2 Register changes to b0010, see *Peripheral Identification Registers* on page 3-31.
- The number of valid bits in value of the FIFOFULL Level Register increases from 7 to 8, See FIFOFULL Level Register in the *Embedded Trace Macrocell Architecture Specification*.

r2p0-r2p1 The implementation revision field of the ETMIDR register changes to b0010, see *ETM ID Register* on page 3-27.

Chapter 2
Functional Description

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

2.1 About the functions

Figure 2-1 shows the main functional blocks and clock domains of the macrocell.

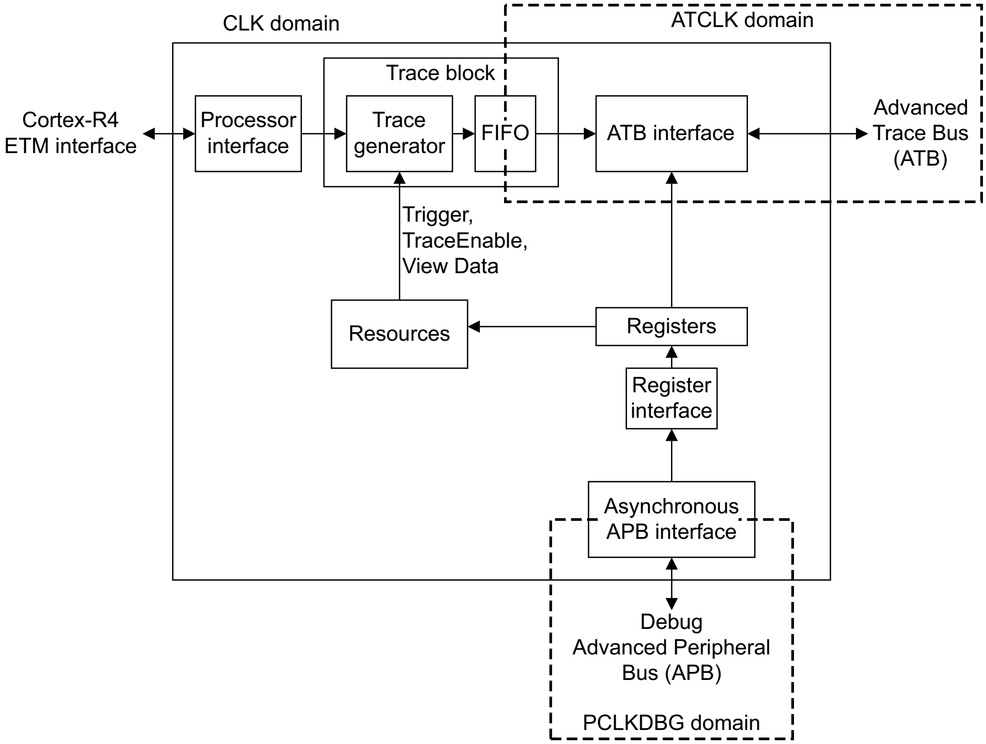


Figure 2-1 ETM-R4 block diagram

2.1.1 Processor interface

This block connects to the Cortex-R4 ETM interface. It pipelines the signals from the processor, decodes the control signals and passes on the information to the internal interfaces.

2.1.2 Trace generator

This block generates the trace packets that are a compressed form of the execution information provided by the Cortex-R4 processor trace generation. The trace packets are then passed to the FIFO.

2.1.3 FIFO

This block buffers bursts of trace packets and manages the transfer of them into the ATCLK domain. Up to 23 bytes of trace packet information can be written into the FIFO in one cycle and four bytes can be read out.

2.1.4 Resources

These blocks contain various comparators and state machines that are programmed by trace software to trigger and filter the trace information. They start and stop trace generation, depending on the conditions that have been set.

2.1.5 ATB Interface

This block reads up to four bytes of packet information from the FIFO and sends them over the ATB interface. It is also responsible for the insertion of flush, alignment synchronization and trigger information into the trace stream.

2.1.6 Asynchronous APB interface

This block implements the interface to the APB, that provides access to the programmable registers. It provides address decoding and pipelining of the address and data to and from the APB. The programmable registers reside in the **CLK**, **ATCLK** and **PCLKDBG** clock domains and so this block manages the synchronization of the access from the APB **PCLKDBG** clock domain to the other two clock domains.

2.2 Interfaces

The ETM-R4 macrocell has the following interfaces:

ATB A 32-bit wide ATB, used for trace output from the macrocell. Up to four bytes of trace packet information can be transferred over the bus in one clock cycle. This interface has hand shaking signals that indicate when trace data is valid and when the receiving component is ready to accept data. There are also signals to request and acknowledge a flush of the trace information and to indicate when a trigger condition has occurred.

See the *AMBA 3 ATB Protocol Specification* for more information about this interface.

APB An APB that provides access to the programmable registers in the ETM-R4 and connects to the system Debug APB. This interface is used to configure the ETM-R4 for a trace session.

See the *AMBA 3 APB Protocol Specification* for more information about this interface.

Processor trace

The Cortex-R4 passes its execution information to ETM-R4 over this interface. This interface is divided into two main sections for instruction and data execution information.

The instruction section contains instruction address and control information. The information carried on the control bus includes:

- the number of instructions executed in the same cycle
- changes in program flow
- the processor instruction state
- condition code evaluation
- exception information.

The data section contains address, data and control information. The address bus carries the addresses of memory locations accessed by load and store instructions. The data bus is 64-bits wide and carries the data values transferred by load, store and coprocessor register transfer instructions. The information carried by the control bus includes the type, direction and size of a data transfer. There is also a context ID bus that indicates the context ID value of the processor.

This interface also includes:

- The 47-bit wide Event bus. See *Interaction with the Performance Monitoring Unit* on page 2-10.
- Debug state request/acknowledge signals.

- Wait for interrupt handshaking signals.
- A signal from the ETM to power up the interface.

Miscellaneous

The ETM-R4 has other interface signals that:

- Configure the ETM. See *Configurable options* on page 1-10.
- Input and output external resource information that controls triggering and filtering of the trace stream.
- Selects the processor that is enabled as the trace source on the processor trace interface.
- Enable invasive and non-invasive debug.

Test This interface contains the scan enable and reset bypass signals used in production testing of the ETM-R4.

2.3 Clocking and resets

The following sections describe the ETM-R4 clocks, clock enables and clock resets:

- *ETM-R4 clock signals*
- *ETM-R4 clock enable signals*
- *ETM-R4 resets*.

2.3.1 ETM-R4 clock signals

ETM-R4 has the following clocks:

CLK	This is the main clock for the ETM-R4 block and must be the same clock as that wired to the **CLK** input of the Cortex-R4 processor. It can be asynchronous to **PCLKDBG** and **ATCLK**.
PCLKDBG	This is the Debug APB interface clock for ETM-R4. It can be asynchronous to **CLK**. The *CoreSight Technology System Design Guide* requires **PCLKDBG** and **ATCLK** to be synchronous.
ATCLK	This is the ATB interface clock. It can be asynchronous to **CLK**. The *CoreSight Technology System Design Guide* requires **PCLKDBG** and **ATCLK** to be synchronous.

Figure 2-1 on page 2-2 shows these clock domains.

Note

Typically, in a SoC, you drive **PCLKDBG** at half the frequency of **ATCLK**.

2.3.2 ETM-R4 clock enable signals

ETM-R4 has the following clock enable signals:

ATCLKEN	This is the ATB clock enable. It can slow down **ATCLK**.
PCLKENDBG	This is the Debug APB interface clock enable. It can slow down **PCLKDBG**.

2.3.3 ETM-R4 resets

ETM-R4 has the following resets:

nSYSPORESET	This signal is the main power-on reset. It resets all the registers in the ETM-R4. It is active LOW.

PRESETDBGn	This signal is the Debug APB interface reset. It resets all the registers in the ETM-R4. It is active LOW.

2.4 Operation

This section describes the implementation-defined features of the operation of the ETM-R4 macrocell. It contains the following sections:

- *Implementation-defined registers*
- *Precise TraceEnable events* on page 2-9
- *Parallel instruction execution* on page 2-9
- *Context ID tracing* on page 2-9
- *Trace and Comparator features* on page 2-9
- *Interaction with the Performance Monitoring Unit* on page 2-10
- *Other implementation-defined features of the macrocell* on page 2-11.

See the *Embedded Trace Macrocell Architecture Specification* for more information about the operation of the ETM-R4 macrocell.

2.4.1 Implementation-defined registers

There are two groups of ETM registers:

- registers that are completely defined by the *Embedded Trace Macrocell Architecture Specification*
- registers that are at least partly implementation-defined.

Chapter 3 *Programmers Model* gives more information about the ETM registers in:

- *Register summary* on page 3-6
- *Register descriptions* on page 3-18.

In Chapter 3, the following sections describe each of the implementation-defined registers:

- *ETM Main Control Register* on page 3-18
- *Configuration Code Register* on page 3-23
- *ASIC Control Register* on page 3-26
- *ETM ID Register* on page 3-27
- *Configuration Code Register* on page 3-23
- *Extended External Input Selection Register* on page 3-30
- *Power-Down Status Register* on page 3-31
- The Integration test registers:
  - *Processor-ETM Interface Register* on page 3-40
  - *Miscellaneous Outputs Register* on page 3-42
  - *Miscellaneous Inputs Register* on page 3-43
  - *Trigger Acknowledge Register* on page 3-44

  - *Trigger Request Register* on page 3-45
  - *ATB Data Register 0* on page 3-46
  - *ATB Control Register 0* on page 3-49
  - *ATB Control Register 1* on page 3-48
  - *ATB Control Register 2* on page 3-47.
- *Peripheral Identification Registers* on page 3-31
- *Component Identification Registers* on page 3-35.

2.4.2 Precise TraceEnable events

The *Embedded Trace Macrocell Architecture Specification* states that **TraceEnable** is imprecise under certain conditions, with some implementation-defined exceptions. When the enabling event selects the following resources, it does not cause **TraceEnable** to be imprecise, provided that the resources are themselves precise:

- single address comparators
- address range comparators.

2.4.3 Parallel instruction execution

The Cortex-R4 processor supports parallel instruction execution. This means the macrocell is capable of tracing two instructions per cycle.

Although the trace start/stop block is evaluated for each instruction as required, the macrocell cannot trace one instruction without the other. In other words, if one instruction is traced, the instruction it is paired with is also always traced. If **ViewData** is active, any data associated with the paired instruction is also traced.

2.4.4 Context ID tracing

The macrocell detects the `MCR` instruction that changes the context ID, and traces the appropriate number of bytes as a context ID packet instead of a normal data packet. This means that if context ID tracing is enabled, an `MCR` instruction that changes the context ID does not have its data traced separately.

2.4.5 Trace and Comparator features

In ETM Architecture v3.3, it is implementation-defined whether an ETM supports a number of Trace and Comparator features. This section specifies the implementation of these features on the CoreSight ETM-R4 macrocell:

- *Trace features* on page 2-10
- *Comparator features* on page 2-10.

Trace features

ETM-R4 implements all of the ETMv3.3 trace features. This means it supports:

- data value and data address tracing
- data suppression
- cycle-accurate tracing.

For descriptions of these features see the *Embedded Trace Macrocell Architecture Specification*.

Comparator features

ETM-R4 implements data address comparison on the CoreSight ETM-R4 macrocell. For a description of data address comparison see the *Embedded Trace Macrocell Architecture Specification*.

2.4.6 Interaction with the Performance Monitoring Unit

The Cortex-R4 processor includes a PMU that enables events, such as cache misses and instructions executed, to be counted over a period of time. The macrocell can still use these events by means of the extended external input facility. Each bit in the **EVNTBUS[46:0]** input is mapped to the corresponding extended external input. See the *Cortex-R4 Technical Reference Manual* for more information on the mapping of events to bits within this bus.

Some events use two bits. Two of these events can occur in a cycle. They must be dealt with separately if they are to be properly counted.

The Cortex-R4 PMU can count the two external outputs as additional events. These events are not provided back to the macrocell as extended external inputs.

These facilities enable additional filtering of the system events using ETM resources, such as instruction address ranges or the start/stop resource, before they are passed back to the PMU for counting. To do this:

- Configure the ETM extended external input selectors to the system events you want to count.
- Configure the required ETM filtering resource as appropriate.
- Configure the ETM external outputs to extended external input selector and the required ETM filtering resource.
- Select the ETM external outputs as the events to be counted in the Cortex-R4 PMU.

2.4.7 Other implementation-defined features of the macrocell

The following implementation-defined features of the macrocell do not affect the descriptions of the features given in the *Embedded Trace Macrocell Architecture Specification*:

- Bits[1:0] of the Synchronization Frequency Register are reserved, RAZ, WI.
- Value Not Traced packets are not output in data-only mode. When data address tracing is enabled in data-only mode, an address packet is output for each traced data transfer that is not sequential to the data address of the previously traced data transfer.

2.5 Constraints and limitations of use

This section describes the constraints and limitations of use that apply to the ETM-R4 macrocell. It contains the following sections:

- *Trace limitations*
- *PortMode and PortSize*.

2.5.1 Trace limitations

There are no trace limitations.

2.5.2 PortMode and PortSize

The macrocell only supports a 32-bit port size, and only supports the dynamic port mode.

In the ETMCR, at offset `0x0`, from reset:

- the PortSize bits, bits[21, 6:4], take the value b0100, indicating a 32-bit port
- the PortMode bits, bits[13, 17:16], take the value b000, indicating dynamic port mode.

For more information see *ETM Main Control Register* on page 3-18.

Chapter 3
Programmers Model

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

3.1 About the programmers model

This chapter describes the mechanisms for programming the registers used to set up the trace and triggering facilities of the macrocell. The programmers model enables you to use the ETM registers to control the macrocell.

The following sections describe the programmers model:

- *Controlling ETM programming* on page 3-3
- *Programming and reading ETM registers* on page 3-4
- *Register summary* on page 3-6
- *Register descriptions* on page 3-18.

3.2 Mode of operation

The following sections describes how you control ETM programming.

3.2.1 Controlling ETM programming

When programming the ETM registers you must enable all the changes at the same time. For example, if the counter is reprogrammed, it might start to count based on incorrect events, before the trigger condition is correctly set up.

You can use the ETM programming bit in the ETMCR to disable all trace operations during programming. See *ETM Main Control Register* on page 3-18. To do this follow the procedure shown in Figure 3-1.

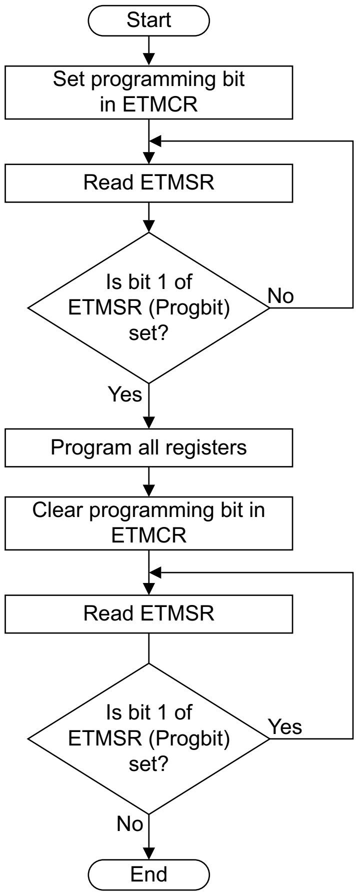


Figure 3-1 Programming ETM registers

The processor does not have to be in the debug state while you program the ETM registers.

3.2.2 Programming and reading ETM registers

You program and read the ETM registers using the Debug APB interface. This provides a direct method of programming:

- a stand-alone macrocell
- a macrocell in a CoreSight system.

3.3 Data structures

See the *Embedded Trace Macrocell Architecture Specification* for descriptions of the trace packet formats generated by the ETM-R4 macrocell.

3.4 Register summary

This section summarizes the ETM registers. For full descriptions of the ETM registers, see:

- *Register descriptions* on page 3-18, for the implementation-defined registers.
- The *Embedded Trace Macrocell Architecture Specification*, for the other registers.

Table 3-1 on page 3-7 lists all of the registers, and tells you describe each register. The registers are listed in register number order.

The macrocell registers are listed by functional group in the section *Functional grouping of registers* on page 3-12. The functional group register tables include additional information about each register:

- The register access type. This is read-only, write-only or read/write.
- The clock domain of the register.
- The base offset address of the register. The base address of a register is always four times its register number.
- Additional information about the implementation of the register, where appropriate.

Note

- Registers not listed here are not implemented. Reading a non-implemented register address returns 0. Writing to a non-implemented register address has no effect.
- In Table 3-1 on page 3-7:
  - The Default value column shows the value of the register immediately after an ETM reset. For read-only registers, every read of the register returns this value.
  - The listed Functional group table gives more information about the register, including its clock domain.
  - Access type is described as follows:

RW	Read and write
RO	Read only
WO	Write only.

All ETM registers are 32-bits wide.

Table 3-1 ETM-R4 register summary

Register number	Name	Type	Reset	Group[a]	Description
`0x000`	ETMCR	RW	`0x00000441`	1	*ETM Main Control Register* on page 3-18
`0x001`	ETMCCR	RO	`0x8D014024`[b]	1	*Configuration Code Register* on page 3-23
`0x002`	ETMTRIGGER	RW	-[c]	4	*Trigger Event Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x003`	ETMASICCTLR	RW	`0x00000000`	1	*ASIC Control Register* on page 3-26
`0x004`	ETMSR,	RW	-[c]	1	*ETM Status Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x005`	ETMSCR	RO	`0x00020C0C`[d]	1	*System Configuration Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x006`	ETMTSSCR	RW	-[c]	2	*TraceEnable Start/Stop Control Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x007`	ETMTECR2	RW	-[c]	2	*TraceEnable Control 2 Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x008`	ETMTEEVR	RW	-[c]	2	*TraceEnable Event Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x009`	ETMTECR1	RW	-[c]	2	*TraceEnable Control 1 Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x00B`	ETMFFLR[e]	RW	-[c]	1	*FIFOFULL Level Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x00C`	ETMVDEVR	RW	-[c]	2	*ViewData Event Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x00D`	ETMVDCR1	RW	-[c]	2	*ViewData Control 1 Register* in the *Embedded Trace Macrocell Architecture Specification*

Table 3-1 ETM-R4 register summary (continued)

Register number	Name	Type	Reset	Group[a]	Description
`0x00F`	ETMVDCR3	RW	-[c]	2	*ViewData Control 3 Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x010` to `0x017`	ETMACVR1-8	RW	-[c]	3	*Address Comparator Value Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x020` to `0x027`	ETMACTR1-8	RW	-[c]	3	*Address Comparator Access Type Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x030`[f]	ETMDCVR1[f]	RW	-[c]	3	*Data Comparator Value Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x032`[f]	ETMDCVR3[f]	RW	-[c]	3	*Data Comparator Value Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x040`[f]	ETMDCMR1[f]	RW	-[c]	3	*Data Comparator Mask Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x042`[f]	ETMDCMR3[f]	RW	-[c]	3	*Data Comparator Mask Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x050`, `0x051`	ETMCNTRLDVR1-2	RW	-[c]	4	*Counter Reload Value Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x054`, `0x055`	ETMCNTENR1-2	RW	-[c]	4	*Counter Enable Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x058`, `0x059`	ETMCNTRLDEVR1-2	RW	-[c]	4	*Counter Reload Event Registers in the Embedded Trace Macrocell Architecture Specification*
`0x05C`, `0x05D`	ETMCNTVR1-2	RW	-[c]	4	*Counter Value Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x060` to `0x065`	ETMSQEVR	RW	-[c]	4	*Sequencer State Transition Event Registers* in the *Embedded Trace Macrocell Architecture Specification*

Table 3-1 ETM-R4 register summary (continued)

Register number	Name	Type	Reset	Group[a]	Description
`0x067`	ETMSQR	RW	-[c]	4	*Current Sequencer State Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x068`, `0x069`	ETMEXTOUTEVR1-2	RW	-[c]	4	*External Output Event Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x06C`	ETMCIDCVR	RW	-[c]	3	*Context ID Comparator Value Registers* in the *Embedded Trace Macrocell Architecture Specification*
`0x06F`	ETMCIDCMR	RW	-[c]	3	*Context ID Comparator Mask Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x078`	ETMSYNCFR	RW	`0x00000400`	1	*Synchronization Frequency Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x079`	ETMIDR	RO	`0x4104F23x`[g]	1	*ETM ID Register* on page 3-27
`0x07A`	ETMCCER	RO	`0x0000097A`	1	*Configuration Code Extension Register* on page 3-28
`0x07B`	ETMEXTINSELR	RW	-[c]	4	*Extended External Input Selection Register* on page 3-30
`0x080`	ETMTRACEIDR	RW	`0x00000000`	1	*CoreSight Trace ID Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x0C5`	ETMPDSR	RO	-[c]	1	*Power-Down Status Register* on page 3-31
`0x3B6`	ITETMIF	RO[h]	-[i]	6	*Processor-ETM Interface Register* on page 3-40
`0x3B7`	ITMISCOUT	WO	n/a[j]	6	*Miscellaneous Outputs Register* on page 3-42
`0x3B8`	ITMISCIN	RO[h]	-[i]	6	*Miscellaneous Inputs Register* on page 3-43
`0x3B9`	ITTRIGGERACK	RO[h]	-[i]	6	*Trigger Acknowledge Register* on page 3-44
`0x3BA`	ITTRIGGERREQ	WO	n/a[j]	6	*Trigger Request Register* on page 3-45

Table 3-1 ETM-R4 register summary (continued)

Register number	Name	Type	Reset	Group[a]	Description
`0x3BB`	ITATBDATA0	WO	n/a[j]	6	*ATB Data Register 0* on page 3-46
`0x3BC`	ITATBCTR2	RO[h]	-[i]	6	*ATB Control Register 2* on page 3-47
`0x3BD`	ITATBCTR1	WO	n/a[j]	6	*ATB Control Register 1* on page 3-48
`0x3BE`	ITATBCTR0	WO	n/a[j]	6	*ATB Control Register 0* on page 3-49
`0x3C0`	ETMITCTRL	RW	`0x00000000`	5	*Integration Mode Control Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x3E8`	ETMCLAIMSET	RW	`0x000000FF`	5	*Claim Tag Set Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x3E9`	ETMCLAIMCLR	RW	`0x00000000`	5	*Claim Tag Clear Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x3EC`	ETMLAR	WO	n/a[j]	5	*Lock Access Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x3ED`	ETMLSR	RO	-[i]	5	*Lock Status Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x3EE`	ETMAUTHSTATUS	RO	-[i]	5	*Authentication Status Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x3F2`	ETMDEVID	RO	`0x00000000`	5	*CoreSight Device Configuration Register* in the *Embedded Trace Macrocell Architecture Specification*
`0x3F3`	ETMDEVTYPE	RO	`0x00000013`	5	*CoreSight Device Type Register* in the *Embedded Trace Macrocell Architecture Specification*

Table 3-1 ETM-R4 register summary (continued)

Register number	Name	Type	Reset	Group[a]	Description
`0x3F4` to `0x3F7`	ETMPIDR4-7	RO	-[i]	5	*Peripheral Identification Registers* on page 3-31
`0x3F8` to `0x3FB`	EETMPIDR0-3	RO	-[i]	-	-
`0x3FC` to `0x3FF`	ETMCIDR0-3	RO	-[i]	5	*Component Identification Registers* on page 3-35

a. Functional group. For more information, see:
   for Group 1, *General control and ID registers* on page 3-12, Table 3-2 on page 3-12
   for Group 2, *TraceEnable and ViewData registers* on page 3-13, Table 3-3 on page 3-13
   for Group 3, *Comparator registers* on page 3-14, Table 3-4 on page 3-14
   for Group 4, *Counter, Sequencer and other resource registers* on page 3-15, Table 3-5 on page 3-15
   for Group 5, *CoreSight Management registers* on page 3-16, Table 3-6 on page 3-16
   for Group 6, *Integration Test registers* on page 3-17, Table 3-7 on page 3-17.

b. Default value when **MAXEXTOUT[1:0]** and **MAXEXTIN[2:0]** are all tied LOW (0), see the register description for more information.

c. The register is not reset by a reset of the macrocell. Therefore, it does not have a specific default value, and its reset value is Unknown.

d. Bits[14:12] of the System Configuration Register are tied to the **MAXCORES[2:0]** signals. If a **MAXCORES** bit is High then the corresponding bit in the System Configuration Register is set to 1, for example if **MAXCORES[0]** is tied HIGH then bit[12] is set to 1. The default value given is for all **MAXCORES** signals tied LOW, bits[14:12] = b000.
   For more information about the **MAXCORES[2:0]** signals, see *ETM-R4 Signals* on page A-2.

e. Although the macrocell does not include **FIFOFULL** logic, the FIFOFULL Level Register controls the FIFO level where data suppression occurs. For more information see the *Embedded Trace Macrocell Architecture Specification*.

f. In the Data Comparator register area, even number registers are reserved. For the CoreSight ETM-R4, reserved areas are:
   Register `0x031`, Data Comparator Value 1, at offset `0x0C4`   Register `0x033`, Data Comparator Value 3, at offset `0x0CC`
   Register `0x041`, Data Comparator Mask 1, at offset `0x104`   Register `0x043`, Data Comparator Mask 3, at offset `0x10C`.
   You must not write to these reserved register addresses. Reads from these addresses are Unpredictable.

g. The value of bits[3:0] of the ETMIDR depend on the macrocell revision, see *ETM ID Register* on page 3-27 for more information.

h. The values of the read-only Integration Test registers are valid only when the macrocell is in Integration Test mode. If you read one of these registers when the macrocell is in normal operating mode the result returned is Unknown.

i. See the register description for more information.

j. Not applicable. These are write-only registers.

3.4.1 Functional grouping of registers

This section lists the macrocell registers by functional group, as follows:

- *General control and ID registers*
- *TraceEnable and ViewData registers* on page 3-13
- *Comparator registers* on page 3-14
- *Counter, Sequencer and other resource registers* on page 3-15
- *CoreSight Management registers* on page 3-16
- *Integration Test registers* on page 3-17.

These functional groups include all of the registers.

General control and ID registers

Table 3-2 lists the general control and ID registers in register number order.

Table 3-2 General control and ID registers

Register number	Name	Base offset	Clock domain	Description
`0x000`	ETM Control	`0x000`	**CLK**	*ETM Main Control Register* on page 3-18
`0x001`	Configuration Code	`0x004`	**CLK**	*Configuration Code Register* on page 3-23
`0x003`	ASIC Control	`0x00C`	**CLK**	*ASIC Control Register* on page 3-26
`0x004`	ETM Status	`0x010`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x005`	System Configuration	`0x014`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x00B`	FIFOFULL Level[a]	`0x02C`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x078`	Synchronization Frequency	`0x1E0`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.[b]
`0x079`	ETM ID	`0x1E4`	**CLK**	*ETM ID Register* on page 3-27
`0x07A`	Configuration Code Extension	`0x1E8`	**CLK**	*Configuration Code Extension Register* on page 3-28
`0x080`	CoreSight Trace ID	`0x200`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x0C5`	Power-Down Status	`0x314`	**CLK**	*Power-Down Status Register* on page 3-31

a. Although the macrocell does not include **FIFOFULL** logic, the FIFOFULL Level Register controls the FIFO level where data suppression occurs. For more information see the *Embedded Trace Macrocell Architecture Specification*.
b. Only bits[11:2] of the Synchronization Frequency Register are implemented. Bits[1:0] Read-As-Zero.

TraceEnable and ViewData registers

Table 3-3 lists the TraceEnable and ViewData registers in register number order.

Table 3-3 TraceEnable and ViewData registers

Register number	Name	Base offset	Clock domain	Description
`0x006`	TraceEnable Start/Stop Resource control	`0x018`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x007`	TraceEnable Control 2	`0x01C`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x008`	TraceEnable Event	`0x020`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x009`	TraceEnable Control 1	`0x024`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x00C`	ViewData Event	`0x030`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x00D`	ViewData Control 1	`0x034`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x00F`	ViewData Control 3	`0x03C`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.

Comparator registers

Table 3-4 lists the Comparator registers in register number order. These control the Address, Data and Context ID comparators.

Table 3-4 Comparator registers

Register number	Name	Base offset	Clock domain	Description
`0x010` to `0x017`	Address Comparator Value 1-8	`0x040` to `0x05F`	**CLK**	*Embedded Trace Macrocell Architecture Specification*
`0x020` to `0x027`	Address Comparator Access Type 1-8	`0x080` to `0x09F`	**CLK**	*Embedded Trace Macrocell Architecture Specification*[a]
`0x030`[b]	Data Comparator Value 1[b]	`0x0C0`[b]	**CLK**	*Embedded Trace Macrocell Architecture Specification*
`0x032`[b]	Data Comparator Value 3[b]	`0x0C8`[b]	**CLK**	*Embedded Trace Macrocell Architecture Specification*
`0x040`[b]	Data Comparator Mask 1[b]	`0x100`[b]	**CLK**	*Embedded Trace Macrocell Architecture Specification*
`0x042`[b]	Data Comparator Mask 3[b]	`0x108`[b]	**CLK**	*Embedded Trace Macrocell Architecture Specification*
`0x06C`	Context ID Comparator Value	`0x1B0`	**CLK**	*Embedded Trace Macrocell Architecture Specification*
`0x06F`	Context ID Comparator Mask	`0x1BC`	**CLK**	*Embedded Trace Macrocell Architecture Specification*

[a] Because the Cortex-R4 processor does not implement the Security Extensions, only bits[9:0] of the Address Comparator Access Type Registers are implemented.

[b] In the Data Comparator register area, even number registers are reserved. For the CoreSight ETM-R4, reserved areas are:
Register `0x031`, Data Comparator Value 1, at offset `0x0C4` Register `0x033`, Data Comparator Value 3, at offset `0x0CC`
Register `0x041`, Data Comparator Mask 1, at offset `0x104` Register `0x043`, Data Comparator Mask 3, at offset `0x10C`.
You must not write to these reserved register addresses. The value of a reads from these addresses is Unknown.

Counter, Sequencer and other resource registers

Table 3-5 lists the Counter, Sequencer and other resource registers in register number order. These control:

- the two Counters, and associated events
- the Sequencer, and associated state change events
- Trigger events
- EXTOUT (External Output) events
- Extended External Input selection.

Table 3-5 Counter, Sequencer and other resource registers

Register number	Name	Base offset	Clock domain	Description
`0x002`	Trigger Event	`0x008`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x050`, `0x051`	Counter Reload Value 1-2	`0x140`, `0x144`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x054`, `0x055`	Counter Enable Event 1-2	`0x150`, `0x154`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x058`, `0x059`	Counter Reload Event 1-2	`0x160`, `0x164`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x05C`, `0x05D`	Counter Value 1-2	`0x170`, `0x174`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x060` to `0x065`	Sequencer State Transition Events	`0x180` to `0x194`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x067`	Current Sequencer State	`0x19C`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x068`, `0x069`	External Output Event 1-2	`0x1A0`, `0x1A4`	**CLK**	*Embedded Trace Macrocell Architecture Specification*.
`0x07B`	Extended External Input Selector	`0x1EC`	**CLK**	*Extended External Input Selection Register* on page 3-30.

CoreSight Management registers

Table 3-6 lists the CoreSight Management registers in register number order.

Table 3-6 CoreSight Management registers

Register number	Name	Base offset	Clock domain	Description
`0x3C0`	Integration Mode Control	`0xF00`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3E8`	Claim Tag Set	`0xFA0`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3E9`	Claim Tag Clear	`0xFA4`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3EC`	Lock Access	`0xFB0`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3ED`	Lock Status	`0xFB4`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3EE`	Authentication Status	`0xFB8`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3F2`	Device Configuration	`0xFC8`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3F3`	Device Type	`0xFCC`	**PCLKDBG**	*Embedded Trace Macrocell Architecture Specification*
`0x3F4` to `0x3F7`	Peripheral ID4 to 7	`0xFD0` to `0xFDC`	**PCLKDBG**	*Peripheral Identification Registers* on page 3-31
`0x3F8` to `0x3FB`	Peripheral ID0 to 3	`0xFE0` to `0xFEC`	**PCLKDBG**	
`0x3FC` to `0x3FF`	Component ID0 to 3	`0xFF0` to `0xFFC`	**PCLKDBG**	*Component Identification Registers* on page 3-35

Integration Test registers

Table 3-7 lists the Integration Test registers in register number order.

Table 3-7 Integration Test registers

Register number	Name	Base offset	Clock domain	Description
`0x3B6`	ITETMIF	`0xED8`	**CLK**	*Processor-ETM Interface Register* on page 3-40
`0x3B7`	ITMISCOUT	`0xEDC`	**CLK**	*Miscellaneous Outputs Register* on page 3-42
`0x3B8`	ITMISCIN	`0xEE0`	**CLK**	*Miscellaneous Inputs Register* on page 3-43
`0x3B9`	ITTRIGGERACK	`0xEE4`	**ATCLK**	*Trigger Acknowledge Register* on page 3-44
`0x3BA`	ITTRIGGERREQ	`0xEE8`	**ATCLK**	*Trigger Request Register* on page 3-45
`0x3BB`	ITATBDATA0	`0xEEC`	**ATCLK**	*ATB Data Register 0* on page 3-46
`0x3BC`	ITATBCTR2	`0xEF0`	**ATCLK**	*ATB Control Register 2* on page 3-47
`0x3BD`	ITATBCTR1	`0xEF4`	**ATCLK**	*ATB Control Register 1* on page 3-48
`0x3BE`	ITATBCTR0	`0xEF8`	**ATCLK**	*ATB Control Register 0* on page 3-49

3.5 Register descriptions

The following sections describe the implementation-defined CoreSight ETM-R4 registers:

- *ETM Main Control Register*
- *Configuration Code Register* on page 3-23
- *ASIC Control Register* on page 3-26
- *ETM ID Register* on page 3-27
- *Configuration Code Extension Register* on page 3-28
- *Extended External Input Selection Register* on page 3-30
- *Power-Down Status Register* on page 3-31
- *Peripheral Identification Registers* on page 3-31
- *Component Identification Registers* on page 3-35
- *Integration Test Registers* on page 3-37.

The *Embedded Trace Macrocell Architecture Specification* describes the other CoreSight ETM-R4 registers.

3.5.1 ETM Main Control Register

The ETMCR characteristics are:

Purpose	Controls general operation of the ETM, such as whether tracing is enabled or coprocessor data is traced.
Usage constraints	There are no usage constraints.
Configurations	Always available.
Attributes	See the register summary in Table 3-1 on page 3-7 and Table 3-2 on page 3-12.

Figure 3-2 on page 3-19 shows the ETMCR bit assignments.

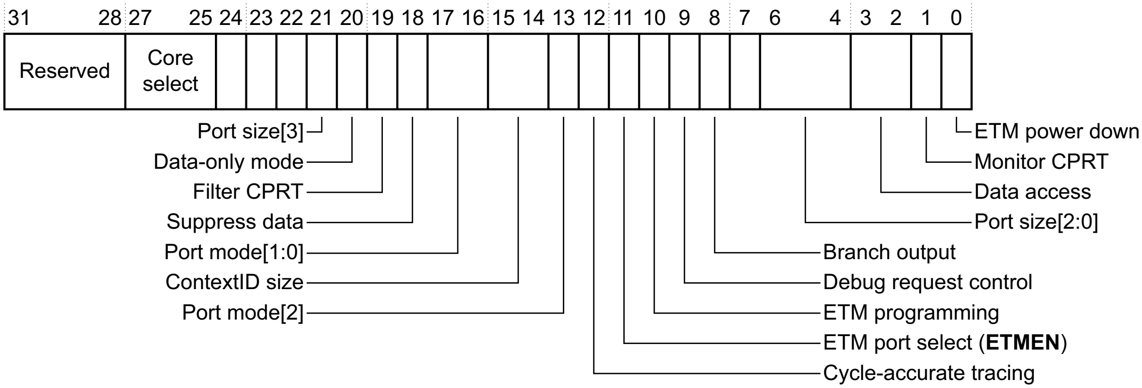


Figure 3-2 ETMCR bit assignments

Table 3-8 lists the ETMCR bit assignments.

Table 3-8 ETMCR bit assignments

Bit	Function	Access	Description
[31:28]	Reserved	RW	Must be written as 0.
[27:25]	Core select	RW	If an ETM is shared between multiple processors, selects the processor to trace. For the maximum value permitted, see bits [14:12] of the ETMSCR bit assignments on page 3-152. To guarantee that the ETM is correctly synchronized to the new processor, you must update these bits as follows: 1. Set bit [10], ETM programming, to `b1`. 2. Poll bit [1] of the ETM Status Register until it is set to `b1`, as described in *Controlling ETM programming* on page 3-3. 3. Set bit [0], ETM power down, to `b1`. 4. Change the Processor select bits. 5. Clear bit [0], ETM power down, to `b0`. 6. Perform other programming required as normal. On an ETM reset these bits are all zero.
[24]	Instrumentation resources access control	RO	ETM-R4 does not implement any instrumentation resources and therefore this bit is RAZ.
[23]	Disable software writes	RO	ETM-R4 does not support this feature and therefore this bit is RAZ.

Table 3-8 ETMCR bit assignments (continued)

Bit	Function	Access	Description
[22]	Disable register writes from the debugger	RO	ETM-R4 does not support this feature and therefore this bit is RAZ.
[21]	Port size[3]	RW	Use this bit in conjunction with bits[6:4]. On an ETM reset this bit is 0, corresponding to the 32-bit port size.
[20]	Data-only mode	RW	The possible values of this bit are: **0** Instruction trace enabled. **1** Instruction trace disabled. Data-only tracing is possible in this mode. On an ETM reset this bit is 0.
[19]	Filter (CPRT)	RW	Use this bit in conjunction with bit[1], the MonitorCPRT bit. See Filter Coprocessor Register Transfers (CPRT) in ETMv3.0 and later in the *Embedded Trace Macrocell Architecture Specification* for more information. On an ETM reset this bit is 0.
[18]	Suppress data	RW	Use this bit with bit[7] to suppress data. See Data suppression in the *Embedded Trace Macrocell Architecture Specification* for more information. On an ETM reset this bit is 0.
[17:16]	Port mode[1:0]	RW	These bits are used, in conjunction with bit[13], to set the trace port clocking mode. ETM-R4 supports only dynamic mode, corresponding to the value b000, but you can write other values to these bits, and a read of the register returns the value written. Writing another value to these bits has no effect on the ETM. Bit[11] of the System Configuration Register indicates if these bits are set to select a supported clocking mode. On an ETM reset these bits are zero. For more information about trace port clocking modes see the *Embedded Trace Macrocell Architecture Specification*.

Table 3-8 ETMCR bit assignments (continued)

Bit	Function	Access	Description
[15:14]	Context ID size	RW	The possible values of this field are: **b00** No Context ID tracing. **b01** Context ID bits[7:0] traced. **b10** Context ID bits[15:0] traced. **b11** Context ID bits[31:0] traced. **Note** Only the number of bytes specified are traced even if the new value is larger than this. On an ETM reset this field is zero.
[13]	Port mode[2]	RW	See the description of bits[17:16]. On an ETM reset this bit is 0.
[12]	Cycle-accurate tracing	RW	Set this bit to 1 if you want the trace to include a precise cycle count of executed instructions. This is achieved by adding extra information into the trace, giving cycle counts even when **TraceEnable** is inactive. On an ETM reset this bit is 0.
[11]	ETM port selection	RW	This bit controls an external output, **ETMEN**. The possible values are: **0** **ETMEN** is LOW. **1** **ETMEN** is HIGH. You can use the **ETMEN** signal to control the routing of trace port signals to shared GPIO pins on your SoC, under the control of logic external to the ETM. Trace software tools must set this bit to 1 to ensure that trace output is enabled from this ETM. On an ETM reset this bit is 0.
[10]	ETM programming	RW	When set to 1, the ETM is being programmed. For more information, see ETM Programming bit and associated state in the *Embedded Trace Macrocell Architecture Specification*. On an ETM reset this bit is set to b1.
[9]	Debug request control	RW	If you set this bit to 1, when the trigger event occurs, the **DBGRQ** output is asserted until **DBGACK** is observed. This enables the ARM processor to be forced into Debug state. On an ETM reset this bit is 0.

Table 3-8 ETMCR bit assignments (continued)

Bit	Function	Access	Description
[8]	Branch output	RW	Set this bit to 1 if you want the ETM to output all branch addresses, even if the branch is because of a direct branch instruction. Setting this bit to 1 enables reconstruction of the program flow without having access to the memory image of the code being executed. On an ETM reset this bit is 0.
[7]	Stall processor	RO	ETM-R4 does not implement **FIFOFULL** stalling of the processor, and therefore this bit is RAZ.
[6:4]	Port size[2:0]	RW	Use this field with bit[21] to specify the port size. The port size determines how many external pins are available to output the trace information on **ATDATA[31:0]**. ETM-R4 supports only the 32-bit port size, corresponding to a Port size[3:0] value of b0100, but you can write other values to these bits, and a read of the register returns the value written. Writing another value to these bits has no effect on the ETM. Bit[10] of the System Configuration Register indicates if these bits are set to select an unsupported port size. For more information see the *Embedded Trace Macrocell Architecture Specification*. On an ETM reset this field is b100, corresponding to the 32-bit port size.
[3:2]	Data access	RW	This field configures the data tracing mode. The possible values are: **b00** No data tracing. **b01** Trace only the data portion of the access. **b10** Trace only the address portion of the access. **b11** Trace both the address and the data of the access. On an ETM reset this field is zero.
[1]	MonitorCPRT	RW	This field controls whether CPRTs are traced. The possible values are: **0** CPRTs not traced. **1** CPRTs traced. This bit is used with bit[19]. See Filter Coprocessor Register Transfers (CPRT) in ETMv3.0 and later in the *Embedded Trace Macrocell Architecture Specification* for more information. On an ETM reset this bit is 0.

Table 3-8 ETMCR bit assignments (continued)

Bit	Function	Access	Description
[0]	ETM power down	RW	A pin controlled by this bit enables the ETM power to be controlled externally, see *Control of ETM power down*. The sense of this bit is inverted, and drives the **ETMPWRUP** signal. This bit must be cleared by the trace software tools at the beginning of a debug session. When this bit is set to 1, ETM tracing is disabled and accesses to any registers other than this register and the Lock Access Register are ignored. On an ETM reset this bit is set to 1. See *Control of ETM power down* for additional information on controlling ETM power down.

Control of ETM power down

You can use the **ETMPWRUP** signal, controlled by the ETM power down bit of the ETMCR, to gate the clock to the logic in the ETM interface of the processor, to save power. Also, when you set the ETM power down bit to 1, the clock to most of the logic in the ETM is gated, disabling ETM tracing and leaving the ETM block operating in a low-power mode.

Note

You must not use the **ETMEN** signal to gate the ETM clock or any other functionality required for basic operation. You can use the **ETMEN** signal to control functionality that is required only for off-chip tracing, such as multiplexing between two ETMs. Use the **ETMPWRUP** signal to control basic operation of the ETM.

3.5.2 Configuration Code Register

The ETMCCR characteristics are:

Purpose	Indicates the configuration of the ETM-R4 macrocell.
Usage constraints	There are no usage constraints.
Configurations	Always available.
Attributes	See the register summary in Table 3-1 on page 3-7 and Table 3-2 on page 3-12. If the **MAXEXTOUT[1:0]** and **MAXEXTIN[2:0]** signals are all tied LOW (0) the ETMCCR has the value `0x8D014024`.

Figure 3-3 shows the ETMCCR bit assignments.

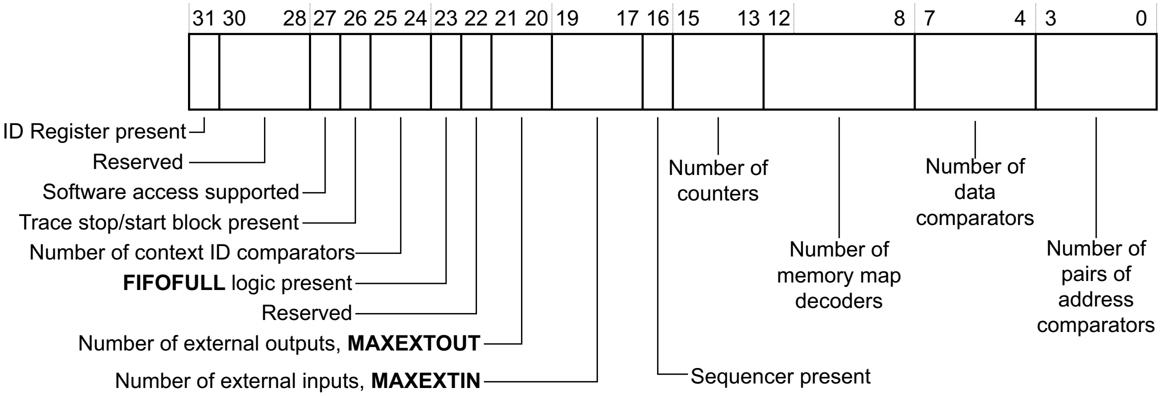


Figure 3-3 ETMCCR bit assignments

Table 3-9 lists the ETMCCR bit assignments

Table 3-9 ETMCCR bit assignments

Bits	Value	Function
[31]	1	ETMIDR present.
[30:28]	b000	Reserved. *Read-As-Zero* (RAZ).
[27]	1	Software access is supported.
[26]	1	Trace start/stop block is present.
[25:24]	b01	Number of Context ID comparators.
[23]	0	**FIFOFULL** logic absent.
[22]	0	Reserved, Read-As-Zero. The *Embedded Trace Macrocell Architecture Specification* defines this as the most significant bit of the Number of external outputs field, see the description of bits[21:20].
[21:20]	-	Number of external outputs. Determined by the **MAXEXTOUT[1:0]** inputs. The maximum value of this field is 2, because CoreSight ETM-R4 supports a maximum of 2 external outputs.

Table 3-9 ETMCCR bit assignments (continued)

Bits	Value	Function
[19:17]	-	Number of external inputs. Determined by the **MAXEXTIN[2:0]** inputs. The maximum value of this field is 4, because CoreSight ETM-R4 supports a maximum of 4 external inputs.
[16]	1	The sequencer is present.
[15:13]	2	Number of counters.
[12:8]	0	Number of memory map decoders.
[7:4]	2	Number of data comparators.
[3:0]	4	Number of pairs of address comparators.

3.5.3 ASIC Control Register

The ETMASICCR characteristics are:

Purpose	Controls the **ASICCTL[7:0]** signal.
Usage constraints	There are no usage constraints.
Configurations	Always available.
Attributes	See the register summary in Table 3-1 on page 3-7 and Table 3-2 on page 3-12.

Figure 3-4 shows the ETMASICCR bit assignments.

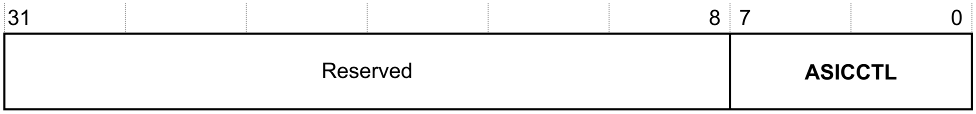


Figure 3-4 ETMASICCR bit assignments

Table 3-10 lists the ETMASICCR bit assignments.

Table 3-10 ETMASICCR bit assignments

Bits	Function
[31:8]	Reserved.
[7:0]	**ASICCTL**[7:0]: When a bit in this field is set to 0 the corresponding bit of **ASICCTL[7:0]** is LOW. When a bit in this field is set to 1 the corresponding bit of **ASICCTL[7:0]** is HIGH. On an ETM reset, these bits are 0.

3.5.4 ETM ID Register

The ETMIDR characteristics are:

Purpose	Identifies the implementation of ETM-R4.
Usage constraints	There are no usage constraints.
Configurations	Always available.
Attributes	This register has the value `0x4104F23x`, where x depends on the release version of the macrocell, see the Implementation revision field description in *ETMIDR bit assignments* for more information. See the register summary in Table 3-1 on page 3-7 and Table 3-2 on page 3-12.

Figure 3-5 shows the ETMIDR bit assignments.

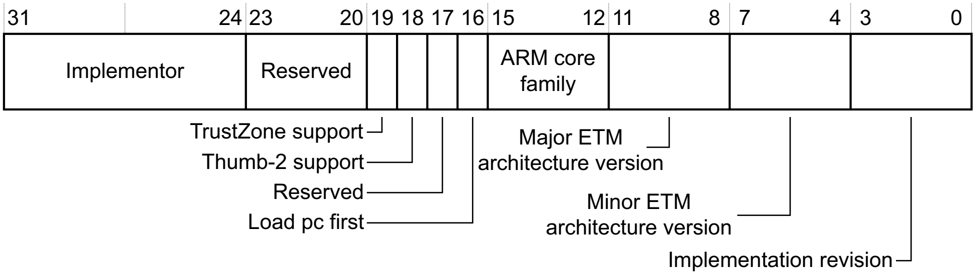


Figure 3-5 ETMIDR bit assignments

Table 3-11 lists the ETMIDR bit assignments.

Table 3-11 ETMIDR bit assignments

Bit numbers	Value	Function
[31:24]	0x41	Implementer = A (for ARM).
[23:20]	b0000	Reserved.
[19]	0	Security Extensions support. This bit is set to 1 if the processor supports the ARMv7 architecture Security Extensions. On the macrocell, this bit is not set (=0), meaning that the ETM behaves as if the processor is in Secure state at all times.

Table 3-11 ETMIDR bit assignments (continued)

Bit numbers	Value	Function
[18]	`1`	Thumb-2 support. This bit is set to 1 if the processor supports the Thumb-2 architectural extensions. On the macrocell, this bit is set to 1, meaning that all 32-bit Thumb instructions are traced as a single instruction, including BL and BLX immediate.
[17]	`0`	Reserved.
[16]	`0`	If set to 1, load PC first. On the macrocell, this bit is not set (=0), meaning that on an LSM[a] load operation with the PC included in the load list, the PC is *not* loaded first.
[15:12]	`b1111`	ARM processor family. The value of b1111 means that the processor family is defined elsewhere.
[11:8]	`b0010`	Major ETM architecture version number. A value of 0 in this field indicates ETMv1. For ETMv3.x, this field = 2.
[7:4]	`b0011`	Minor ETM architecture version number. For ETMvx.3, this field = 3.
[3:0]	`b0011`	Implementation revision. Value given is for the r2p1 release of the macrocell. For release r0p0 the value is `b0000`. For release r1p0 the value is `b0001`. For release r2p0 the value is `b0010`.

[a] See the *Embedded Trace Macrocell Architecture Specification* for a definition and list of LSM operations.

3.5.5 Configuration Code Extension Register

The ETMCCER characteristics are:

Purpose	Indicates the configuration of the extended external input bus.
Usage constraints	There are no usage constraints.
Configurations	Always available.
Attributes	See the register summary in Table 3-1 on page 3-7 and Table 3-2 on page 3-12.

Figure 3-6 on page 3-29 shows the ETMCCER bit assignments.

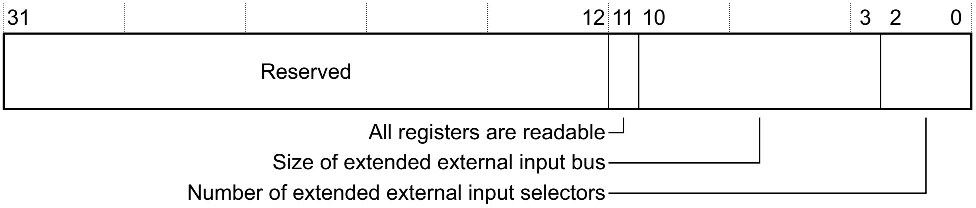


Figure 3-6 ETMCCER bit assignments

Table 3-12 lists the ETMCCER bit assignments.

Table 3-12 ETMCCER bit assignments

Bit numbers	Value	Function
[31:12]	0	Reserved, RAZ.
[11]	1	All registers, except some integration test registers, are readable. See Table 3-7 on page 3-17 for more information on the access to integration test registers[a].
[10:3]	47	Size of extended external input bus.
[2:0]	b010	Number of extended external input selectors.

[a] Registers with names that start with IT are the Integration Test Registers, for example ITATBCTR1.

3.5.6 Extended External Input Selection Register

The ETMEXTINSELR characteristics are:

Purpose	Specifies the extended external inputs, see the *Embedded Trace Macrocell Architecture Specification* for more information.
Usage constraints	There are no usage constraints.
Configurations	Always available.
Attributes	See the register summary in Table 3-1 on page 3-7 and Table 3-5 on page 3-15.

Figure 3-7 shows the ETMEXTINSELR bit assignments.

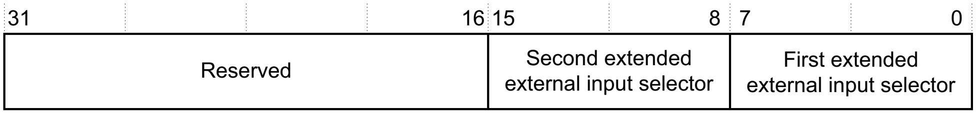


Figure 3-7 ETMEXTINSELR bit assignments

Table 3-13 lists the ETMEXTINSELR bit assignments.

Table 3-13 ETMEXTINSELR bit assignments

Bits	Description
[31:16]	Reserved, SBZP.
[15:8]	Second extended external input selector: **Bits[15,14]** Reserved, SBZP. **Bits[13:8]** Selection value for second external input.
[7:0]	First extended external input selector: **Bits[7,6]** Reserved, SBZP. **Bits[5:0]** Selection value for first external input.

3.5.7 Power-Down Status Register

The ETMPDSR characteristics are:

Purpose	Indicates the power-down status of the ETM.
Usage constraints	There are no usage constraints.
Configurations	Always available.
Attributes	See the register summary in Table 3-1 on page 3-7 and Table 3-2 on page 3-12.

Figure 3-8 shows the ETMPDSR bit assignments.

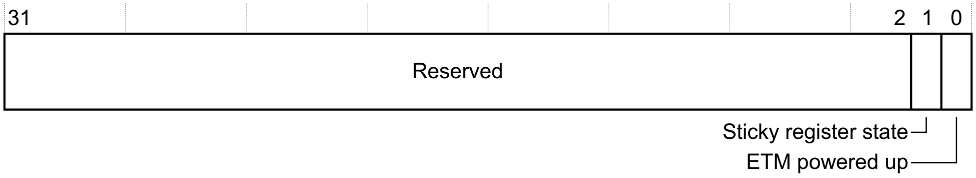


Figure 3-8 ETMPDSR bit assignments

Table 3-14 lists the ETMPDSR bit assignments.

Table 3-14 ETMPDSR bit assignments

Bit numbers	Value	Function
[31:2]	0	Reserved, RAZ.
[1]	0	Sticky Register State. ETM-R4 does not support multiple power domains so this bit is RAZ.
[0]	1	ETM Powered Up. The ETM Trace registers are accessible. ETM-R4 does not support multiple power domains so this bit is RAO.

3.5.8 Peripheral Identification Registers

The ETMPIDR0-ETMPIDR7 characteristics are:

Purpose	Provides the standard Peripheral ID required by all CoreSight components, see the *Embedded Trace Macrocell Architecture Specification* for more information
Usage constraints	Only bits[7:0] of each register are used. This means that ETMPIDR0-ETMPIDR7 define a single 64-bit *Peripheral ID*, as Figure 3-9 on page 3-32 shows.

Configurations Always available.

Attributes See the register summary in Table 3-1 on page 3-7 and Table 3-6 on page 3-16.

Figure 3-9 shows the mapping between ETMPIDR0-ETMPIDR7 and the single 64-bit *Peripheral ID* value,

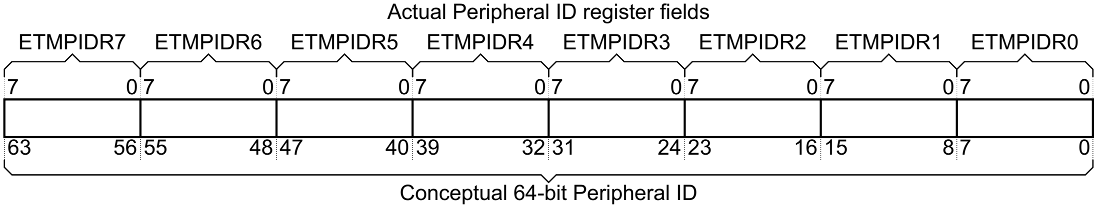
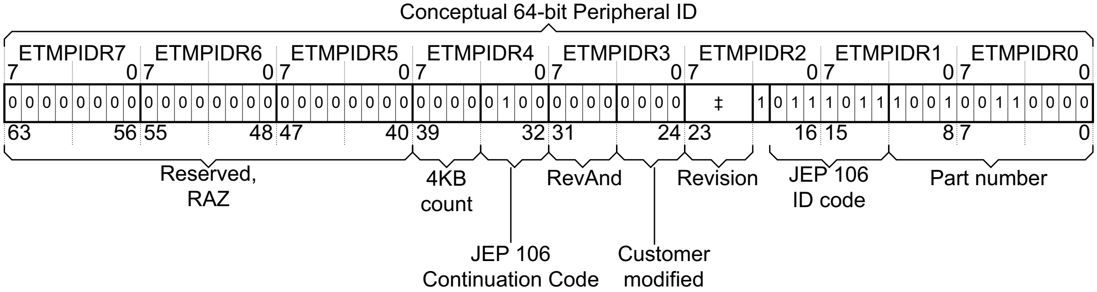


Figure 3-9 Mapping between ETMPIDR0-ETMPIDR7 and the Peripheral ID value

Figure 3-10 shows the Peripheral ID bit assignments in the single conceptual Peripheral ID register.



‡ See text for the value of the Revision field

Figure 3-10 Peripheral ID fields

Table 3-15 lists the values of the fields when reading this set of registers. The *Embedded Trace Macrocell Architecture Specification* gives more information about many of these fields.

Table 3-15 ETMPIDR0-ETMPIDR7 bit assignments

Register	Register number	Register offset	Bit	Value	Description
ETMPIDR7	`0x3F7`	`0xFDC`	[31:8]	-	Unused, read undefined.
			[7:0]	`0x00`	Reserved for future use, RAZ.
ETMPIDR6	`0x3F6`	`0xFD8`	[31:8]	-	Unused, read undefined.
			[7:0]	`0x00`	Reserved for future use, RAZ.
ETMPIDR5	`0x3F5`	`0xFD4`	[31:8]	-	Unused, read undefined.
			[7:0]	`0x00`	Reserved for future use, RAZ.
ETMPIDR4	`0x3F4`	`0xFD0`	[31:8]	-	Unused, read undefined.
			[7:4]	`0x0`	n, where 2^n is number of 4KB blocks used.
			[3:0]	`0x4`	JEP 106 continuation code.
ETMPIDR3	`0x3FB`	`0xFEC`	[31:8]	-	Unused, read undefined.
			[7:4]	`0x0`	RevAnd (at top level). Manufacturer revision number.
			[3:0]	`0x0`	Customer Modified. `0x0` indicates from ARM.
ETMPIDR2	`0x3FA`	`0xFE8`	[31:8]	-	Unused, read undefined.
			[7:4]	[a]	Revision Number of Peripheral. This value is the same as the Implementation revision field of the ETMIDR, see *ETM ID Register* on page 3-27.
			[3]	1	Always 1. Indicates that a JEDEC assigned value is used.
			[2:0]	b011	JEP 106 identity code[6:4].

Table 3-15 ETMPIDR0-ETMPIDR7 bit assignments (continued)

Register	Register number	Register offset	Bit	Value	Description
ETMPIDR1	`0x3F9`	`0xFE4`	[31:8]	-	Unused, read undefined.
			[7:4]	`b0001`	JEP 106 identity code[3:0].
			[3:0]	`0x9`	Part Number[11:8]. Upper *Binary Coded Decimal* (BCD) value of Device Number.
ETMPIDR0	`0x3F8`	`0xFE0`	[31:8]	-	Unused, read undefined.
			[7:0]	`0x30`	Part Number[7:0]. Middle and Lower BCD value of Device Number.

a. See the Description column for more information.

Note

In Table 3-15 on page 3-33, the *Peripheral Identification Registers* on page 3-31 are listed in order of register name, from most significant (ETMPIDR7) to least significant (ETMPIDR0). This does not match the order of the register offsets. Similarly, in Table 3-16 on page 3-35 the *Component Identification Registers* on page 3-35 are listed in order of register name, from most significant (ETMCIDR3) to least significant (ETMCIDR0).

3.5.9 Component Identification Registers

The ETMCIDR0-ETMCIDR3 characteristics are:

Purpose	Identifies the ETM as a CoreSight component. For more information, see the *Embedded Trace Macrocell Architecture Specification*.
Usage constraints	Only bits[7:0] of each register are used. This means that ETMCIDR0-ETMCIDR3 define a single 32-bit Component ID, as Figure 3-11 shows.
Configurations	Always available.
Attributes	See the register summary in Table 3-1 on page 3-7 and Table 3-6 on page 3-16.

Figure 3-11 shows the mapping between ETMCIDR0-ETMCIDR3 and the single 64-bit *Component ID* value.

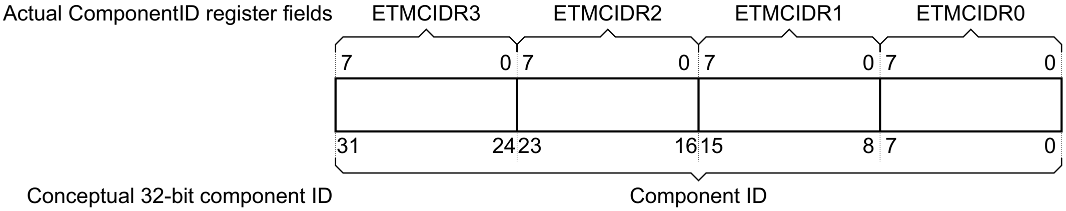


Figure 3-11 Mapping between ETMCIDR0-ETMCIDR3 and the Component ID value

Table 3-16 lists the Component ID bit assignments in the single conceptual Component ID register.

Table 3-16 ETMCIDR0-ETMCIDR3, bit assignments

Register	Register number	Register offset	Bit	Value	Description
ETMCIDR3	0x3FF	`0xFFC`	[31:8]	-	Unused, read undefined.
			[7:0]	`0xB1`	Component identifier, bits[31:24].
ETMCIDR2	0x3FE	`0xFF8`	[31:8]	-	Unused, read undefined.
			[7:0]	`0x05`	Component identifier, bits[23:16].
ETMCIDR1	0x3FD	`0xFF4`	[31:8]	-	Unused, read undefined.

Table 3-16 ETMCIDR0-ETMCIDR3, bit assignments (continued)

Register	Register number	Register offset	Bit	Value	Description
			[7:4]	`0x9`	Component class (component identifier, bits[15:12]).
			[3:0]	`0x0`	Component identifier, bits[11:8].
ETMCIDR0	0x3FC	`0xFF0`	[31:8]	-	Unused, read undefined.
			[7:0]	`0x0D`	Component identifier, bits[7:0].

3.5.10 Integration Test Registers

The following subsections describe the Integration Test Registers. To access these registers you must first set bit[0] of the Integration Mode Control Register (ETMITCTRL) to 1.

- You can use the write-only Integration Test Registers to set the outputs of some of the ETM signals. Table 3-17 lists the signals that can be controlled in this way.
- You can use the read-only Integration Test Registers to read the state of some of the ETM input signals. Table 3-18 on page 3-38 lists the signals that can be read in this way.

See the *Embedded Trace Macrocell Architecture Specification* for more information. ETMITCTRL is described in the *Embedded Trace Macrocell Architecture Specification*.

Table 3-17 Output signals that the Integration Test Registers can control

Signal	Register	Bit	Register description
AFREADY	ITATBCTR0	[1]	See *ATB Control Register 0* on page 3-49
ATBYTES[1:0]	ITATBCTR0	[9:8]	See *ATB Control Register 0* on page 3-49
ATDATA[31, 23, 15, 7, 0]	ITATBDATA0	[4:0]	See *ATB Data Register 0* on page 3-46
ATID[6:0]	ITATBCTR1	[6:0]	See *ATB Control Register 1* on page 3-48
ATVALID	ITATBCTR0	[0]	See *ATB Control Register 0* on page 3-49
ETMDBGRQ	ITMISCOUT	[4]	See *Miscellaneous Outputs Register* on page 3-42
EXTOUT[1:0]	ITMISCOUT	[9:8]	See *Miscellaneous Outputs Register* on page 3-42
nETMWFIREADY	ITMISCOUT	[5]	See *Miscellaneous Outputs Register* on page 3-42
TRIGGER	ITTRIGGERREQ	[0]	See *Trigger Request Register* on page 3-45

Table 3-18 Input signals that the Integration Test Registers can read

Signal	Register	Bit	Register description
AFVALID	ITATBCTR2	[1]	See *ATB Control Register 2* on page 3-47
ATREADY	ITATBCTR2	[0]	See *ATB Control Register 2* on page 3-47
DBGACK	ITMISCIN	[4]	See *Miscellaneous Inputs Register* on page 3-43
ETMCID[31, 0]	ITETMIF	[11:10]	See *Processor-ETM Interface Register* on page 3-40
ETMDA[31, 0]	ITETMIF	[7:6]	See *Processor-ETM Interface Register* on page 3-40
ETMDCTL[11, 0]	ITETMIF	[5:4]	See *Processor-ETM Interface Register* on page 3-40
ETMDD[63, 0]	ITETMIF	[9:8]	See *Processor-ETM Interface Register* on page 3-40
ETMIA[31, 1]	ITETMIF	[3:2]	See *Processor-ETM Interface Register* on page 3-40
ETMICTL[13, 0]	ITETMIF	[1:0]	See *Processor-ETM Interface Register* on page 3-40
ETMWFIPENDING	ITMISCIN	[5]	See *Miscellaneous Inputs Register* on page 3-43
EVNTBUS[46, 28, 0]	ITETMIF	[14:12]	See *Processor-ETM Interface Register* on page 3-40
EXTIN[3:0]	ITMISCIN	[3:0]	See *Miscellaneous Inputs Register* on page 3-43
TRIGGERACK	ITTRIGGERACK	[0]	See *Trigger Acknowledge Register* on page 3-44

Using the Integration Test Registers

The *CoreSight ETM-R4 Integration Manual* gives a full description of the use of the Integration Test Registers to check integration. In brief:

When bit[0] of ETMITCTRL is set to 1:

- Values written to the write-only integration test registers map onto the specified outputs of the macrocell. For example, writing `0x3` to ITMISCOUT[9:8] causes **EXTOUT[1:0]** to take the value `0x3`.
- Values read from the read-only integration test registers correspond to the values of the specified inputs of the macrocell. For example, if you read ITMISCIN[3:0] you obtain the value of **EXTIN[3:0]**.

When bit[0] of ETMITCTRL is set to 0:

- Reading an Integration Test Register returns an Unpredictable value.

- The effect of attempting to write to an Integration Test Register, other than the read-only Integration Test Registers, is Unpredictable.

  Note

  You must not attempt to write to an Integration Test Register unless you have set bit[0] of ETMITCTRL to 1.

See the *Embedded Trace Macrocell Architecture Specification* for more information on ETMITCTRL.

Processor-ETM Interface Register

The ITETMIF characteristics are:

Purpose	Reads the state of the ETM input pins shown in Table 3-19.
Usage constraints	• Available when bit[0] of ETMITCTRL is set to 1. • The value of the register depends on the signals on the input pins when the register is read.
Configurations	Always available.
Attributes	See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-12 shows the ITETMIF bit assignments.

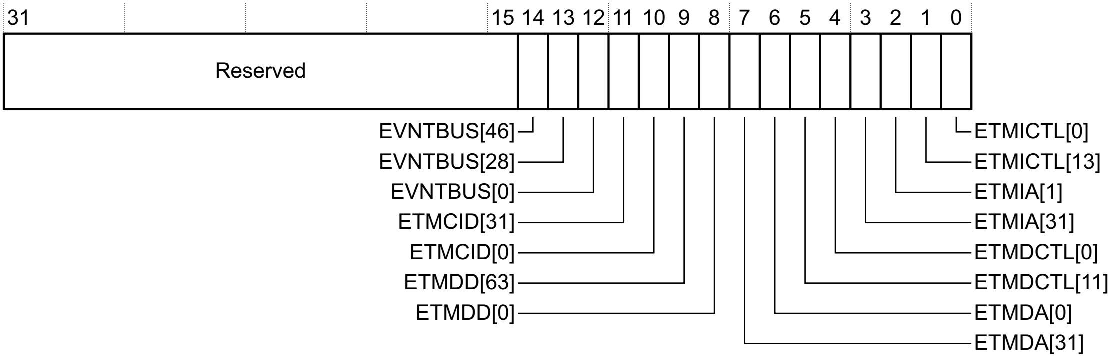


Figure 3-12 ITETMIF bit assignments

Table 3-19 lists the ITETMIF bit assignments.

Table 3-19 ITETMIF bit assignments

Bits	Name	Function
[31:15]	-	Reserved. Read undefined.
[14]	EVNTBUS[46]	Returns the value of the **EVNTBUS[46]** input pin[a].
[13]	EVNTBUS[28]	Returns the value of the **EVNTBUS[28]** input pin[a].
[12]	EVNTBUS[0]	Returns the value of the **EVNTBUS[0]** input pin[a].

Table 3-19 ITETMIF bit assignments (continued)

Bits	Name	Function
[11]	ETMCID[31]	Returns the value of the **ETMCID[31]** input pin[a].
[10]	ETMCID[0]	Returns the value of the **ETMCID[0]** input pin[a].
[9]	ETMDD[63]	Returns the value of the **ETMDD[63]** input pin[a].
[8]	ETMDD[0]	Returns the value of the **ETMDD[0]** input pin[a].
[7]	ETMDA[31]	Returns the value of the **ETMDA[31]** input pin[a].
[6]	ETMDA[0]	Returns the value of the **ETMDA[0]** input pin[a].
[5]	ETMDCTL[11]	Returns the value of the **ETMDCTL[11]** input pin[a].
[4]	ETMDCTL[0]	Returns the value of the **ETMDCTL[0]** input pin[a].
[3]	ETMIA[31]	Returns the value of the **ETMIA[31]** input pin[a].
[2]	ETMIA[1]	Returns the value of the **ETMIA[1]** input pin[a].
[1]	ETMICTL[13]	Returns the value of the **ETMICTL[13]** input pin[a].
[0]	ETMICTL[0]	Returns the value of the **ETMICTL[0]** input pin[a].

a. When a bit is set to 0, the corresponding input pin is LOW.
When a bit is set to 1, the corresponding input pin is HIGH.
The ITETMIF bit values always correspond to the physical state of the input pins.

Miscellaneous Outputs Register

The ITMISCOUT characteristics are:

Purpose	Sets the state of the output pins shown in Table 3-20.
Usage constraints	• Available when bit[0] of ETMITCTRL is set to 1. • The value of the register sets the signals on the output pins when the register is written.
Configurations	Always available.
Attributes	See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-13 shows the ITMISCOUT bit assignments.

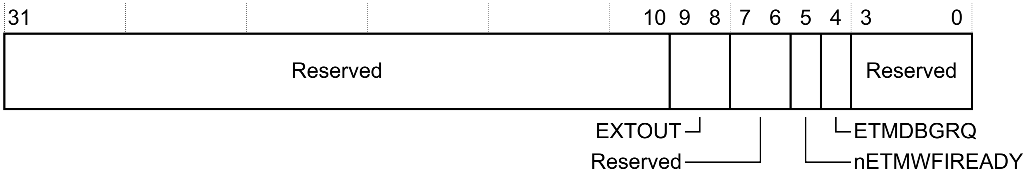


Figure 3-13 ITMISCOUT bit assignments

Table 3-20 lists the ITMISCOUT bit assignments.

Table 3-20 ITMISCOUT bit assignments

Bits	Name	Function
[31:10]	-	Reserved. Write as zero.
[9:8]	EXTOUT	Drives the **EXTOUT[1:0]** output pins[a].
[7:6]	-	Reserved. Write as zero.
[5]	ETMWFIREADY	Drives the **nETMWFIREADY** output pin[a].
[4]	ETMDBGRQ	Drives the **ETMDBGRQ** output pin[a].
[3:0]	-	Reserved. Write as zero.

[a] When an input pin is LOW, the corresponding register bit is 0.
When an input pin is HIGH, the corresponding register bit is 1.
The ITMISCOUT bit values correspond to the physical state of the output pins.

Miscellaneous Inputs Register

The ITMISCIN characteristics are:

Purpose	Reads the state of the input pins shown in Table 3-21.
Usage constraints	• Available when bit[0] of ETMITCTRL is set to 1. • The values of the register bits depend on the signals on the input pins when the register is read.
Configurations	Always available.
Attributes	See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-14 shows the ITMISCIN bit assignments.

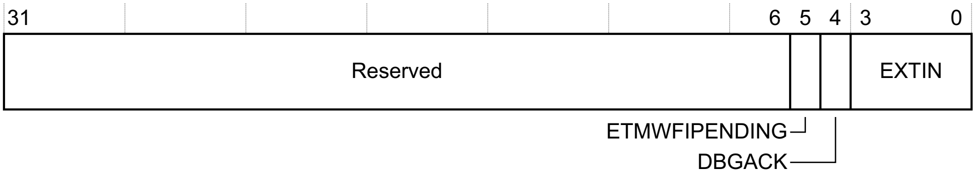


Figure 3-14 ITMISCIN bit assignments

Table 3-21 lists the ITMISCIN bit assignments.

Table 3-21 ITMISCIN bit assignments

Bits	Name	Function
[31:6]	-	Reserved. Read undefined.
[5]	ETMWFIPENDING	Returns the value of the **ETMWFIPENDING** input pin[a].
[4]	DBGACK	Returns the value of the **DBGACK** input pin[a].
[3:0]	EXTIN	Returns the value of the **EXTIN[3:0]** input pins[a].

[a] When an input pin is LOW, the corresponding register bit is 0.
When an input pin is HIGH, the corresponding register bit is 1.
The ITMISCIN bit values always correspond to the physical state of the input pins.

Trigger Acknowledge Register

The ITTRIGGERACK characteristics are:

Purpose	Reads the state of the TRIGGERACK input pin shown in Table 3-22.
Usage constraints	• Available when bit[0] of ETMITCTRL is set to 1. • The values of the register bits depend on the signal on the input pin when the register is read.
Configurations	Always available.
Attributes	See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-15 shows the ITTRIGGERACK bit assignments.

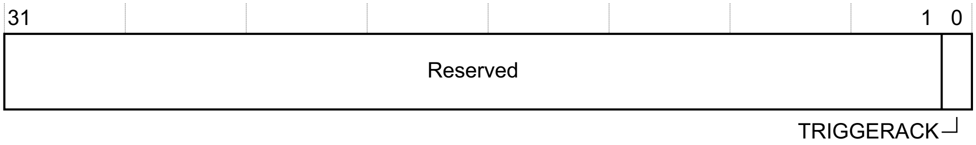


Figure 3-15 ITTRIGGERACK bit assignments

Table 3-22 lists the ITTRIGGERACK bit assignments.

Table 3-22 ITTRIGGERACK bit assignments

Bits	Name	Function
[31:1]	-	Reserved. Read undefined.
[0]	TRIGGERACK	Returns the value of the **TRIGGERACK** input pin[a].

[a] When the TRIGGERACK input pin is LOW, the register bit is 0.
When the TRIGGERACK input pin is HIGH, the register bit is 1.
The ITTRIGGERACK bit value always corresponds to the physical state of the input pin.

Trigger Request Register

The ITTRIGGERREQ characteristics are:

Purpose	Sets the **TRIGGER** output pin shown in Table 3-23.
Usage constraints	• Available when bit[0] of ETMITCTRL is set to 1. • The values of the register bits set the signals on the output pin when the register is written.
Configurations	Always available.
Attributes	See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-16 shows the ITTRIGGERREQ bit assignments.

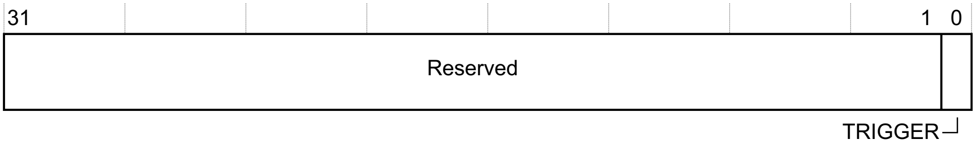


Figure 3-16 ITTRIGGERREQ bit assignments

Table 3-23 lists the ITTRIGGERREQ bit assignments.

Table 3-23 ITTRIGGERREQ bit assignments

Bits	Name	Function
[31:1]	-	Reserved. Write as zero.
[0]	TRIGGER	Drives the **TRIGGER** output pin[a].

[a] When the ITTRIGGERREQ register bit is set to 0, the TRIGGER output pin is LOW.
When the ITTRIGGERREQ register bit is set to 1, the TRIGGER output pin is HIGH.
The ITTRIGGERREQ bit values always correspond to the physical state of the output pins.

ATB Data Register 0

The ITATBDATA0 characteristics are:

Purpose Sets the state of the ATDATA output pins shown in Table 3-24.

Usage constraints
- Available when bit[0] of ETMITCTRL is set to 1.
- The values of the register bits set the signals on the output pins when the register is written.

Configurations Always available.

Attributes See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-17 shows the ITATBDATA0 bit assignments.

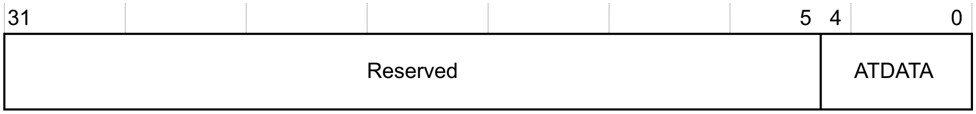


Figure 3-17 ITATBDATA0 bit assignments

Table 3-24 lists the ITATBDATA0 bit assignments.

Table 3-24 ITATBDATA0 bit assignments

Bits	Name	Function
[31:5]	-	Reserved. Write as zero.
[4:0]	ATDATA	Drives the **ATDATA[31, 23, 15, 7, 0]** output pins[a].

[a] When a bit is set to 0, the corresponding output pin is LOW.
When a bit is set to 1, the corresponding output pin is HIGH.
The ITATBDATA0 bit values always correspond to the physical state of the output pins.

ATB Control Register 2

The ITATBCTR2 characteristics are:

Purpose	Reads the state of the **AFVALID** and **ATREADY** input pins from the ATB bus, as shown in Table 3-25.
Usage constraints	• Available when bit[0] of ETMITCTRL is set to 1. • The values of the register bits depend on the signals on the input pins when the register is read.
Configurations	Always available.
Attributes	See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-18 shows the ITATBCTR2 bit assignments.

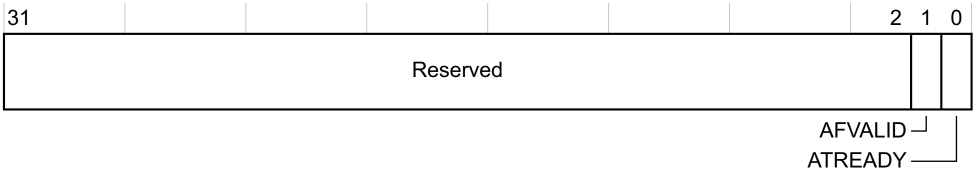


Figure 3-18 ITATBCTR2 bit assignments

Table 3-25 lists the ITATBCTR2 bit assignments.

Table 3-25 ITATBCTR2 bit assignments

Bits	Name	Function
[31:2]	-	Reserved. Read undefined.
[1]	AFVALID	Returns the value of the **AFVALID** input pin[a].
[0]	ATREADY	Returns the value of the **ATREADY** input pin[a].

[a] When an input pin is LOW, the corresponding register bit is 0.
When an input pin is HIGH, the corresponding register bit is 1.
The ITATBCTR2 bit values always correspond to the physical state of the input pins.

ATB Control Register 1

The ITATBCTR1 characteristics are:

Purpose	Sets the state of the **ATID** output pins shown in Table 3-26.
Usage constraints	• Available when bit[0] of ETMITCTRL is set to 1. • The values of the register bits set the signals on the output pins when the register is written.
Configurations	Always available.
Attributes	See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-19 shows the ITATBCTR1 bit assignments.

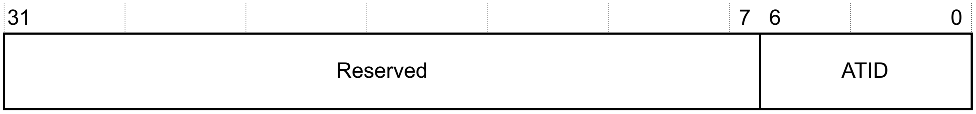


Figure 3-19 ITATBCTR1 bit assignments

Table 3-26 lists the ITATBCTR1 bit assignments.

Table 3-26 ITATBCTR1 bit assignments

Bits	Name	Function
[31:7]	-	Reserved. Write as zero.
[6:0]	ATID	Drives the **ATID[6:0]** output pins[a].

[a] When a bit is set to 0, the corresponding output pin is LOW.
When a bit is set to 1, the corresponding output pin is HIGH.
The ITATBCTR1 bit values always correspond to the physical state of the output pins.

ATB Control Register 0

The ITATBCTR0 characteristics are:

Purpose Sets the state of the output pins shown in Table 3-27.

Usage constraints
- Available when bit[0] of ETMITCTRL is set to 1.
- The values of the register bits set the signals on the output pins when the register is written.

Configurations Always available.

Attributes See the register summaries in Table 3-1 on page 3-7, Table 3-7 on page 3-17 and Table 3-18 on page 3-38.

Figure 3-20 shows the ITATBCTR0 bit assignments.

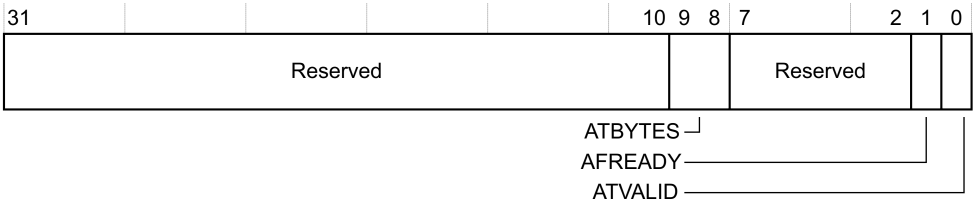


Figure 3-20 ITATBCTR0 bit assignments

Table 3-27 lists the ITATBCTR0 bit assignments.

Table 3-27 ITATBCTR0 bit assignments

Bits	Name	Function
[31:10]	-	Reserved. Write as zero.
[9:8]	ATBYTES	Drives the **ATBYTES[1:0]** output pins[a].
[7:2]	-	Reserved. Write as zero.
[1]	AFREADY	Drives the **AFREADY** output pin[a].
[0]	ATVALID	Drives the **ATVALID** output pin[a].

[a] When a bit is set to 0, the corresponding output pin is LOW. When a bit is set to 1, the corresponding output pin is HIGH. The ITATBCTR0 bit values always correspond to the physical state of the output pins.

Appendix A
Signal Descriptions

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

A.1 ETM-R4 Signals

Table A-1 lists the ETM-R4 signals in alphabetical order. The following sections show the signal directions and the clock domains, for each of the interfaces.

See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-1 ETM-R4 signals

Signal	Description
AFREADY	*ATB interface* on page A-9
AFVALID	*ATB interface* on page A-9
ASICCTL[7:0]	*Miscellaneous interface* on page A-11
ATBYTES[1:0]	*ATB interface* on page A-9
ATCLK	*Clocks and resets* on page A-5
ATCLKEN	*Clocks and resets* on page A-5
ATDATA[31:0]	*ATB interface* on page A-9
ATID[6:0]	*ATB interface* on page A-9
ATREADY	*ATB interface* on page A-9
ATVALID	*ATB interface* on page A-9
CLK	*Clocks and resets* on page A-5
CORESELECT[2:0]	*Miscellaneous interface* on page A-11
DBGACK	*Processor trace interface* on page A-6
DBGEN	*Miscellaneous interface* on page A-11
ETMCID[31:0]	*Processor trace interface* on page A-6
ETMDA[31:0]	*Processor trace interface* on page A-6
ETMDBGRQ	*Processor trace interface* on page A-6
ETMDCTL[11:0]	*Processor trace interface* on page A-6
ETMDD[63:0]	*Processor trace interface* on page A-6
ETMEN	*ATB interface* on page A-9

Table A-1 ETM-R4 signals (continued)

Signal	Description
ETMIA[31:1]	*Processor trace interface* on page A-6
ETMICTL[13:0]	*Processor trace interface* on page A-6
ETMPWRUP	*Processor trace interface* on page A-6
ETMWFIPENDING	*Processor trace interface* on page A-6
EVNTBUS[46:0]	*Processor trace interface* on page A-6
EXTIN[3:0]	*Miscellaneous interface* on page A-11
EXTOUT[1:0]	*Miscellaneous interface* on page A-11
FIFOPEEK[6:0]	*Miscellaneous interface* on page A-11
MAXCORES[2:0]	*Miscellaneous interface* on page A-11
MAXEXTIN[2:0]	*Miscellaneous interface* on page A-11
MAXEXTOUT[1:0]	*Miscellaneous interface* on page A-11
nETMWFIREADY	*Processor trace interface* on page A-6
NIDEN	*Miscellaneous interface* on page A-11
PADDRDBG[11:2]	*APB interface* on page A-8
PADDRDBG31	*APB interface* on page A-8
PCLKDBG	*Clocks and resets* on page A-5
PCLKENDBG	*Clocks and resets* on page A-5
PENABLEDBG	*APB interface* on page A-8
PRDATADBG[31:0]	*APB interface* on page A-8
PREADYDBG	*APB interface* on page A-8
PRESETDBGn	*Clocks and resets* on page A-5
PSELDBG	*APB interface* on page A-8
PWDATADBG[31:0]	*APB interface* on page A-8
PWRITEDBG	*APB interface* on page A-8

Table A-1 ETM-R4 signals (continued)

Signal	Description
RSTBYPASS	*Test interface* on page A-12
SE	*Test interface* on page A-12
nSYSPORESET	*Clocks and resets* on page A-5
TRIGGER	*The trigger signals* on page A-10
TRIGGERACK	*The trigger signals* on page A-10
TRIGSBYPASS	*The trigger signals* on page A-10

A.2 Clocks and resets

Table A-2 lists the clock and reset signals. Input signals must be generated and output signals must be sampled with the clock specified by the clock domain. See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-2 Clock and reset signals

Signal	Direction	Description	Clock domain
CLK	Input	This is the main clock for the ETM-R4.	-
ATCLK	Input	ATB interface clock.	-
ATCLKEN	Input	Enable signal for **ATCLK**.	**ATCLK**
PCLKDBG	Input	Debug APB clock.	-
PCLKENDBG	Input	Debug APB clock enable.	**PCLKDBG**
PRESETDBGn	Input	Debug APB interface reset. Resets all registers.	Internally synchronized
nSYSPORESET	Input	Power-on (main) reset. Resets all registers.	Internally synchronized

A.3 Processor trace interface

Table A-3 lists the trace interface signals from the Cortex-R4. Input signals must be generated and output signals must be sampled with the clock specified by the clock domain. See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-3 Processor trace interface signals

Signal	Direction	Description	Clock domain
ETMICTL[13:0]	Input	Instruction control signals.	**CLK**
ETMIA[31:1]	Input	Address for executed instruction.	**CLK**
ETMDCTL[11:0]	Input	Data control signals.	**CLK**
ETMDA[31:0]	Input	Address for data transfer.	**CLK**
ETMDD[63:0]	Input	Contains the data value for a Load, Store, `MRC`, or `MCR` instruction.	**CLK**
ETMCID[31:0]	Input	Value of the processor Context ID Register.	**CLK**
EVNTBUS[46:0]	Input	Gives the status of the performance monitoring events. Used as extended external inputs.	**CLK**
ETMWFIPENDING	Input	Indicates that the Cortex-R4 processor is about to go into Standby mode, and that the ETM must drain its FIFO.	**CLK**
nETMWFIREADY	Output	Indicates that the macrocell FIFO is empty and that the Cortex-R4 processor can be put into Standby mode.	**CLK**

Table A-3 Processor trace interface signals (continued)

Signal	Direction	Description	Clock domain
ETMDBGRQ	Output	Request from the macrocell for the processor to enter debug state. This must be ORed with any ASIC-level **DBGRQ** signals before being connected to the processor **EDBGRQ** input.	**CLK**
DBGACK	Input	Indicates that the processor is in debug state. This signal is connected to the processor general purpose **DBGACK** output, so that it can be used to determine when **ETMDBGRQ** can be deasserted. It is also used for other purposes in the ETM, and care must be taken to ensure the timing of this signal is appropriate because it does not come through the main interface between the processor and the ETM.	**CLK**
ETMPWRUP	Output	When HIGH, indicates that the macrocell is in use. When LOW: • external logic supporting the macrocell can be clock-gated to conserve power • the Cortex-R4 processor disables the interface • logic within the macrocell is clock-gated to conserve power.	**CLK**

A.4 APB interface

Table A-4 lists the APB signals. Input signals must be generated and output signals must be sampled with the clock specified by the clock domain. See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-4 APB signals

Signal	Direction	Description	Clock domain
PADDRDBG[11:2]	Input	Debug APB Address Bus.	**PCLKDBG**
PADDRDBG31	Input	Originates as an output signal from the *Debug Access Por*t (DAP): • **PADDRDBG31** at logic 1 indicates an access from hardware (JTAG) • **PADDRDBG31** at logic 0 indicates an access from software.	**PCLKDBG**
PENABLEDBG	Input	The Debug APB interface is enabled for a transfer.	**PCLKDBG**
PSELDBG	Input	Debug APB slave select signal.	**PCLKDBG**
PREADYDBG	Output	Used to extend Debug APB transfers.	**PCLKDBG**
PRDATADBG[31:0]	Output	Debug APB read data.	**PCLKDBG**
PWDATADBG[31:0]	Input	Debug APB write data.	**PCLKDBG**
PWRITEDBG	Input	Debug APB transfer direction: 0 = Read 1 = Write.	**PCLKDBG**

A.5 ATB interface

Table A-5 lists the ATB signals. Input signals must be generated and output signals must be sampled with the clock specified by the clock domain. See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-5 ATB signals

Signal	Direction	Description	Clock domain
AFREADY	Output	ATB interface FIFO flush finished.	**ATCLK**
AFVALID	Input	ATB interface FIFO flush request.	**ATCLK**
ATBYTES[1:0]	Output	Size of **ATDATA**.	**ATCLK**
ATDATA[31:0]	Output	ATB interface data.	**ATCLK**
ATID[6:0]	Output	ATB interface trace source ID.	**ATCLK**
ATREADY	Input	**ATDATA** can be accepted.	**ATCLK**
ATVALID	Output	ATB interface data valid.	**ATCLK**
ETMEN	Output	Enable signal for trace output from the ETM, driven by bit[11] of the ETMCR.	**CLK**

A.5.1 The trigger signals

Table A-6 lists the trigger signals. Input signals must be generated and output signals must be sampled with the clock specified by the clock domain. See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-6 Trigger signals

Signal	Direction	Description	Clock domain
TRIGGER	Output	Trigger request status signal.	**ATCLK**
TRIGGERACK	Input	ATB trigger acknowledge.	Internally synchronized
TRIGSBYPASS	Input	Trigger synchronization bypass.	**ATCLK**

The **TRIGSBYPASS**, **TRIGGER**, and **TRIGGERACK** signals control trigger behavior and indicate when a trigger occurs.

TRIGSBYPASS controls whether asynchronous registering and handshaking is performed:

- When TRIGSBYPASS is HIGH, **TRIGGER** is asserted for one **ATCLK** cycle, and **TRIGGERACK** is ignored.
- When TRIGSBYPASS is LOW, **TRIGGER** is asserted, and is held until **TRIGGERACK** is asserted. When **TRIGGERACK** is asserted, **TRIGGER** is de-asserted.

TRIGGERACK is synchronized to the **ATCLK** clock domain inside the ETM, using double registers.

A.6 Miscellaneous interface

Table A-7 lists the ETM sharing signals. Input signals must be generated and output signals must be sampled with the clock specified by the clock domain. See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-7 Miscellaneous signals

Signal	Direction	Description	Clock domain
CORESELECT[2:0]	Output	Where an ETM is shared between multiple processors, this signal specifies the processor to trace. The value appears as bits[14:12] of the System Configuration Register.	**CLK**
MAXCORES[2:0]	Input	Where an ETM is shared between multiple processors, this signal specifies the number of processors the ETM can trace. It must be tied to the number of processors sharing the ETM minus 1. These signals determine the value of bits[14:12] of the System Configuration register, see the footnote to Table 3-1 on page 3-7.	**CLK**
ASICCTL[7:0]	Output	General purpose outputs controlled by the ETMASICCR. See *ASIC Control Register* on page 3-26	**CLK**
DBGEN	Input	Invasive debug enable. When HIGH (1), indicates that invasive debug is enabled.	-
NIDEN	Input	Non-invasive debug enable. When HIGH (1), indicates that non-invasive debug is enabled.	-
EXTIN[3:0]	Input	External input resources.	**CLK**
EXTOUT[1:0]	Output	External outputs.	**CLK**
MAXEXTIN[2:0]	Input	Number of external inputs supported by the ASIC (maximum 4). These signals determine the value bits[19:17] in the ETMCCR, see *Configuration Code Register* on page 3-23.	**CLK**
MAXEXTOUT[1:0]	Input	Number of external outputs supported by the ASIC (maximum 2). These signals determine the value bits[22:20] in the ETMCCR, see *Configuration Code Register* on page 3-23.	**CLK**
FIFOPEEK[6:0]	Output	For validation purposes only. Indicates when various events occur before being written to the FIFO.	**CLK**

A.7 Test interface

Table A-8 lists the scan chain signals. See the *CoreSight ETM-R4 Integration Manual* for information about signals and connectivity.

Table A-8 Test signals

Signal	Direction	Description	Clock domain
RSTBYPASS	Input	Reset synchronization bypass DFT signal.	-
SE	Input	Scan enable DFT signal.	-

Appendix B
AC Characteristics

This appendix describes the macrocell input and output signal timing. It contains the following section:

- *ETM-R4 input and output signal timing parameters* on page B-2.

B.1 ETM-R4 input and output signal timing parameters

Signals are classified according to the percentage of the clock period taken up by internal logic.

- For inputs this is the delay between the input port and the first register.
- For outputs this is the delay between the last register and the output port.

The timing classifications used are based on these delays:

Early	The delay is less than 20% of the period.
Middle	The delay is between 20% and 80% of the period.
Late	The delay is greater than 80% of the period.

Table B-1 describes the ETM-R4 signal timing parameters.

Table B-1 ETM-R4 signal timing parameters

Signal name	Timing classification	Input/Output
AFREADY	Middle	Output
AFVALID	Middle	Input
ASICCTL[7:0]	Middle	Output
ATBYTES[1:0]	Middle	Output
ATCLK	-	Input
ATCLKEN	Middle	Input
ATDATA[31:0]	Middle	Output
ATID[6:0]	Middle	Output
ATREADY	Middle	Input
ATVALID	Middle	Output
CLK	-	Input
CORESELECT[2:0]	Middle	Output
DBGACK	Middle	Input
DBGEN	Middle	Input
ETMCID[31:0]	Middle	Input
ETMDA[31:0]	Middle	Input

Table B-1 ETM-R4 signal timing parameters (continued)

Signal name	Timing classification	Input/Output
ETMDBGRQ	Middle	Output
ETMDD[63:0]	Middle	Input
ETMDCTL[11:0]	Middle	Input
ETMEN	Middle	Output
ETMIA[31:1]	Middle	Input
ETMICTL[13:0]	Middle	Input
ETMPWRUP	Middle	Output
ETMWFIPENDING	Middle	Input
EVNTBUS[46:0]	Middle	Input
EXTIN[3:0]	Middle	Input
EXTOUT[1:0]	Middle	Output
FIFOPEEK[6:0]	Middle	Output
MAXCORES[2:0]	Middle	Input
MAXEXTIN[2:0]	Middle	Input
MAXEXTOUT[1:0]	Middle	Input
nETMWFIREADY	Middle	Output
NIDEN	Middle	Input
PADDRDBG[11:2]	Middle	Input
PADDRDBG31	Middle	Input
PCLKDBG	Middle	Input
PCLKENDBG	Middle	Input
PENABLEDBG	Middle	Input
PRDATADBG[31:0]	Middle	Output
PREADYDBG	Late	Output
PRESETDBGn	Late	Input

Table B-1 ETM-R4 signal timing parameters (continued)

Signal name	Timing classification	Input/Output
PSELDBG	Middle	Input
PWDATADBG[31:0]	Middle	Input
PWRITEDBG	Middle	Input
RSTBYPASS	Middle	Input
SE	Middle	Input
nSYSPORESET	Middle	Input
TRIGGER	Middle	Output
TRIGGERACK	Middle	Input
TRIGSBYPASS	Middle	Input

Note

Actual clock frequencies and input and output timing constraints vary according to application requirements and the silicon process technologies used. The maximum operating clock frequencies change according to the constraints and the process technology you use.

Appendix C
Revisions

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

Table C-1 Differences between issue B and issue C

Change	Location
Changed content in the Introduction	Chapter 1 *Introduction*
Changed the FIFO size from 72 bytes to 144 bytes	Table 1-1 on page 1-6
Changed content in the *Functional Description*, previously *Implementation-defined behavior*	Chapter 2 *Functional Description*
Added reset value to ASIC Control Register	Table 3-1 on page 3-7
Added symbolic names to register summary	Table 3-1 on page 3-7
Changed the revision fields in the ID registers	Table 3-11 on page 3-27

Table C-2 Differences between issue C and issue D

Change	Location	Affects
Changed the revision fields in the ID registers	Table 3-11 on page 3-27	r2p1
Update the JEP 106 continuation code	Table 3-15 on page 3-33	All revisions
Updated description of description of ETM sharing procedure	Table 3-8 on page 3-19	All revisions