CoreSight™ ETM™-A5

Revision: r0p2

Technical Reference Manual

ARM®

ARM DDI 0435C (ID101810)

CoreSight ETM-A5
Technical Reference Manual

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
24 December 2009	A	Non-Confidential	First release for r0p0
06 July 2010	B	Non-Confidential	First release for r0p1
30 September 2010	C	Non-Confidential	First release for r0p2

Proprietary Notice

Confidentiality Status

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

Unrestricted Access is an ARM internal classification.

Product Status

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

Web Address

`http://www.arm.com`

Contents
CoreSight ETM-A5 Technical Reference Manual

List of Tables
CoreSight ETM-A5 Technical Reference Manual

List of Figures
CoreSight ETM-A5 Technical Reference Manual

Preface

This preface introduces the *CoreSight ETM-A5 Technical Reference Manual*. It contains the following sections:

About this book

This book is for the CoreSight *Embedded Trace Macrocell* (ETM) for the Cortex-A5™ and Cortex-A5 MPCore™ processors, the ETM-A5.

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* (ARM IHI 0014).
- Hardware and software engineers integrating the macrocell into an ASIC that includes a Cortex-A5 processor. You can find complementary information in the *CoreSight ETM-A5 Integration Manual* (ARM DIT 0002).

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 all signals.

Appendix B *Revisions*

Read this for a description of technical changes in this document.

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

Conventions

Conventions that this book can use are described in:

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

Typographical

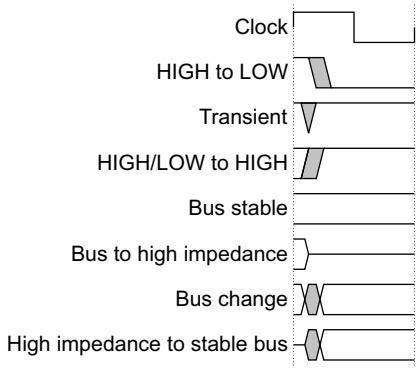
The typographical conventions are:

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.
`mono`space	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.

Further reading

This section lists publications by ARM and by third parties.

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

ARM publications

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

- *CoreSight ETM-A5 Configuration and Sign-off Guide (*ARM DII 0212)
- *CoreSight ETM-A5 Integration Manual* (ARM DIT 0002)
- *CoreSight Design Kit for Cortex-A5 Integration Manual* (ARM DIT 0003)
- *Embedded Trace Macrocell™ Architecture Specification* (ARM IHI 0014)
- *ARM Reference Peripheral Specification* (ARM DDI 0062)
- *Cortex-A5 Technical Reference Manual* (ARM DDI 0433)
- *Cortex-A5 MPCore Technical Reference Manual* (ARM DDI 0434)
- *CoreSight Architecture Specification* (ARM IHI 0020)
- *CoreSight ETM-A5 Release Note* (TM955-DC-06001)
- *AMBA™ 3 ATB Protocol Specification* (ARM IHI 0032)
- *AMBA™ 3 APB Protocol Specification* (ARM IHI 0024).

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 0435C
- 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 CoreSight ETM-A5 macrocell. It contains the following sections:

1.1 About the product

This macrocell provides instruction trace and data trace for the Cortex-A5 microprocessor. The CoreSight ETM-A5 is designed for you to use in a CoreSight system. See the *CoreSight Design Kit A5 Integration Manual* for more information about how a CoreSight system uses the CoreSight ETM-A5 module.

Figure 1-1 shows the main functional blocks of a typical Cortex-A5 system in a *System-on-Chip* (SoC) that includes a CoreSight ETM-A5 macrocell.

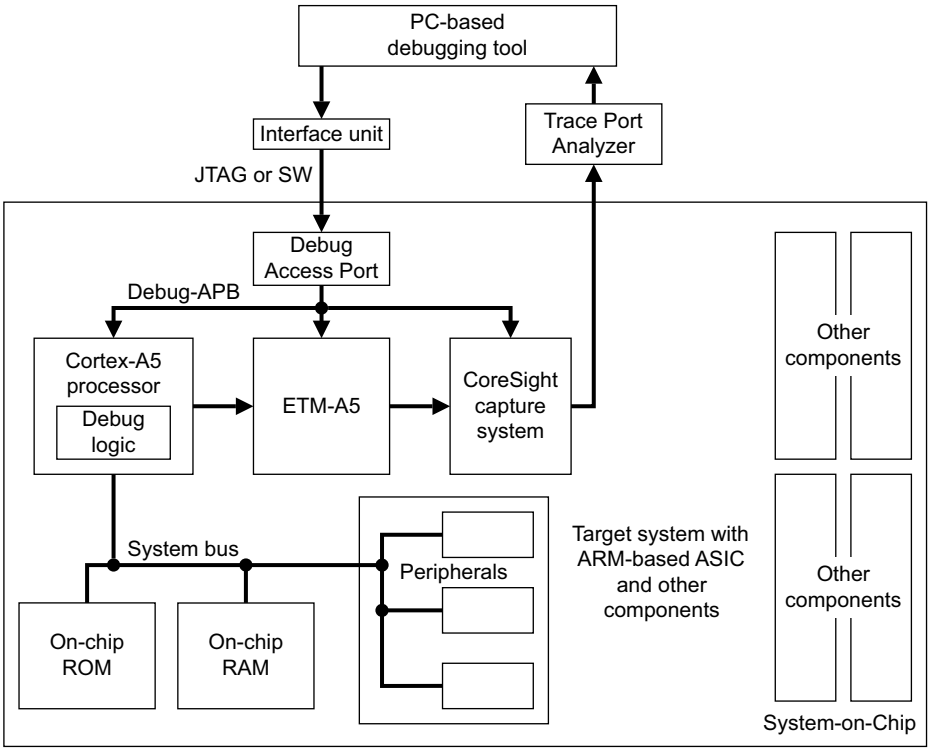


Figure 1-1 Cortex-A5 with ETM-A5 block diagram

1.2 Compliance

ETM-A5 is compatible with the CoreSight architecture, and implements ETM v3.5.

For more information about architectural compliance, see *Architecture and protocol information* on page 1-10.

1.3 Features

ETM-A5 supports tracing of ARM, Thumb, Thumb-EE, and Jazelle instructions.

The *Embedded Trace Macrocell Architecture Specification* describes the features of ETM v3.5.

Table 1-1 lists the features of the ETM-A5 that are implementation-defined, in terms of either:
- the number of times the feature is implemented
- the size of the feature.

Table 1-1 Implementation-specific features of the ETM-A5

Feature	ETM-A5 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	0	See bits [19:16] of the ETMCCER [b]
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	0	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	30	See bits [10:3] of the ETMCCER
Extended external input selectors	2	See bits [2:0] of the ETMCCER [b]
Instrumentation resources	0	See bits [15:13] of the ETMCCER [b]
Trace port size	32-bit	See bits [21,6:4] of the ETMCR [c]
VMID comparator	0	See bit [26] of the ETMCCER [b]
FIFO size	144 bytes	-
ASICCTL general-purpose bus interface	8-bit	See *ASIC Control Register* on page 3-18

a. See *Configuration Code Register* on page 3-17.
b. See *Configuration Code Extension Register* on page 3-20.
c. See *ETM Main Control Register* on page 3-14.

Table 1-2 shows the optional ETM architecture features the ETM-A5 implements.

Table 1-2 ETM-A5 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]

Table 1-2 ETM-A5 implementation of optional features (continued)

Feature	Implemented	Notes
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]
Timestamping	Yes	See bit [28] of the ETMCCER[c]
Data address comparison	Yes	Bit [12] of the ETMCCER[c] reads-as-zero
EmbeddedICE behavior control	No	See bit [21] of the ETMCCER[c]
EmbeddedICE inputs to Trace Start/Stop block	No	See bit [20] of the ETMCCER[c]
Alternative address compression	No	See bit [20] of the ETMIDR
OS Lock mechanism	No	See bits [3, 0] of the ETMOSLSR
Secure non-invasive debug	Yes	See *Embedded Trace Macrocell Architecture Specification*
Context ID tracing	Yes	See bits [15:14] of the ETMCR[b]
VMID tracing	No	See bit [26] of the ETMCCER[c]
Reduced function counter	No	See bit [27] of the ETMCCER[c]

a. See *Configuration Code Register* on page 3-17.
b. See *ETM Main Control Register* on page 3-14.
c. See *Configuration Code Extension Register* on page 3-20.

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

- the trace protocol
- controlling tracing using triggering and filtering resources
- ETM sharing.

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

1.4 Interfaces

A debugger:

- programs the macrocell through its *AMBA Peripheral Bus* (APB) interface. In a CoreSight system, the APB interface connects to the Debug APB bus.
- collects trace data through the *AMBA Trace Bus* (ATB).

The other external connections to the ETM-A5 are:

- 0-4 external inputs
- 0-2 external outputs
- a 30-bit extended external input bus.

See *Configurable options* on page 1-7 for more information about the external inputs and external outputs.

In a SoC the ETM-A5 connects to the Cortex-A5:

- ETM interface.
- *Performance Monitoring Unit* (PMU). The PMU **EVNTBUS** signals connect directly to the ETM-A5 extended external interface bus signals

Interfaces on page 2-3 describes the ETM-A5 interfaces in more detail.

1.5 Configurable options

The ETM-A5 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.
- **TSMAXWIDTH** determines whether the **TSVALUE** input is 64 or 48 bits wide
- **TSNATURAL** determines whether the **TSVALUE** input is Gray coded or natural binary.

On an ETM-A5 implementation, these configuration inputs might be tied off to fixed values, or they might be available as configuration pins on the SoC.

For a description of bits [21:17] of the ETMCCR, see *Configuration Code Register* on page 3-17.

1.6 Test features

The ETM-A5 provides the **DFTSE** and **DFTRSTDISABLE** inputs for testing the implemented device. See the *CoreSight ETM-A5 Integration Manual*.

See also *Integration Test Registers* on page 3-27 for information about the integration test registers provided for testing the ETM-A5 implementation in the SoC.

1.7 Product documentation, design flow, and architecture

This section describes the ETM-A5 books, how they relate to the design flow, and the relevant architectural standards and protocols.

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

1.7.1 Documentation

The ETM-A5 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-A5. It is required at all stages of the design flow. Some behavior described in the TRM might not be relevant because of the way that the ETM-A5 is implemented and integrated. If you are programming the ETM-A5 then contact the integrator to determine the pin configuration of the SoC 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) description with the build configuration options
- 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 documentation from your EDA tools vendor complements the CSG.

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

Integration Manual

The *Integration Manual* (IM) describes how to integrate the ETM-A5 into a SoC. It includes a description of 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-A5.

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

1.7.2 Design flow

The ETM-A5 is delivered as synthesizable RTL. Before it can be used in a product, it must go through the following process:

1. Implementation. The implementer synthesizes the RTL to produce a hard macrocell.
2. Integration. The integrator connects the implemented design into a SoC.
3. Programming. The debug software developer programs the device.

Each stage of the process:

- can be performed by a different party

- can include options that affect the behavior and features at the next stage:

  Build configuration

  The ETM-A5 has no implementer defined options.

  Configuration inputs

  The integrator configures some features of the ETM-A5 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.

  For example, see the **MAXCORES**, **MAXEXTIN**, **MAXEXTOUT**, **TSMAXWIDTH**, and **TSNATURAL** inputs, described in the *CoreSight ETM-A5 Integration Manual*.

  Software configuration

  The programmer configures the ETM-A5 by programming particular values into software-visible registers. This affects the behavior of the ETM-A5.

  See *Register summary* on page 3-5 and *Register descriptions* on page 3-14 for information on the ETM-A5 registers.

Note

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

1.7.3 Architecture and protocol information

The ETM-A5 complies with, or implements, the specifications described in:

- *Embedded Trace Macrocell*
- *CoreSight Architecture*
- *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.

Embedded Trace Macrocell

The ETM-A5 implements ETM architecture version 3.5. See the *Embedded Trace Macrocell Architecture Specification*.

CoreSight Architecture

The ETM-A5 implements CoreSight architecture version 1.0. See the *CoreSight Architecture Specification*.

Advanced Microcontroller Bus Architecture

The ETM-A5 complies with the AMBA 3 protocol. See the *AMBA 3 APB Protocol Specification* and *AMBA 3 ATB Protocol Specification*.

1.8 Product revisions

This section describes the differences in functionality between product revisions:

r0p0 First release.

r0p0-r0p1 No functional changes.

r0p1-r0p2 No functional changes.

Chapter 2
Functional Description

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

2.1 About the functions

Figure 2-1 shows the main functional blocks. The ETM-A5 operates in a single clock domain, **CLK**, which is identical to the processor clock. The ATB and APB interfaces can be operated at slower clock rates, but still synchronously to **CLK**, using the **ATCLKEN** and **PCLKENDBG** clock enables respectively. If the system requires the ETM-A5 to operate in an asynchronous clock domain the appropriate bridge components must be included on the interfaces of the macrocell as shown in Figure 2-1.

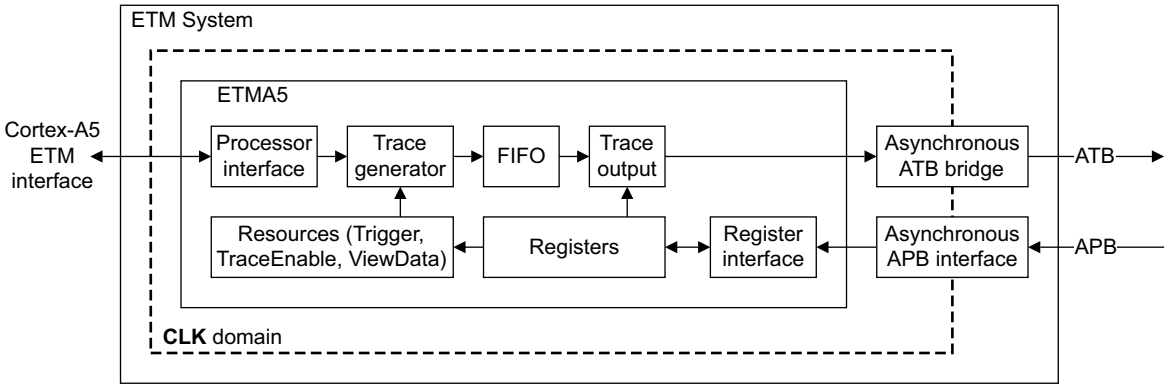


Figure 2-1 Macrocell functional blocks and clock domains

2.2 Interfaces

The ETM-A5 macrocell has the following external interfaces:

ATB A 32-bit ATB, used for trace output from the macrocell. See the *AMBA™ 3 ATB Protocol Specification* for more information about this interface.

APB An APB provides the control interface for the macrocell. See the *AMBA™ 3 APB Protocol Specification* for more information about this interface.

Cortex-A5 ETM interface

The Cortex-A5 processor passes its execution information to the ETM-A5 over this bus. This includes instruction address, branch, exception, data address, data value and Performance Monitoring Unit information.

2.3 Clocking and resets

The following sections describe the ETM-A5 clock, clock enable, and reset signals:

- *Clock signals*
- *Clock enable signals*
- *Resets*.

2.3.1 Clock signals

ETM-A5 uses a single clock, **CLK**. This must be the same clock as that wired to the **CLKIN** input of the Cortex-A5 processor.

2.3.2 Clock enable signals

ETM-A5 has the following clock enable signals:

ATCLKEN This is the ATB clock enable. The ATB can be operated using a slower clock, **ATCLK**, generated by dividing **CLK**. The **ATCLKEN** clock enable must be asserted for one **CLK** cycle on each rising edge of **ATCLK**.

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

2.3.3 Resets

ETM-A5 has the following reset:

nSYSPORESET This signal is the main power-on reset. It is active LOW.

2.4 Operation

This section describes the implementation-defined features of the operation of the ETM-A5. It contains the following sections:

- *Implementation-defined registers*
- *Precise TraceEnable events*
- *Parallel instruction execution* on page 2-6
- *Context ID tracing* on page 2-6
- *Trace and Comparator features* on page 2-6
- *Interaction with the Performance Monitoring Unit (PMU)* on page 2-6
- *Other implementation-defined features of the macrocell* on page 2-7.

See the *Embedded Trace Macrocell Architecture Specification* for more information about the operation of the ETM-A5.

2.4.1 Implementation-defined registers

In terms of how they are defined, 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 the sections:

- *Register summary* on page 3-5
- *Register descriptions* on page 3-14.

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

- *ETM Main Control Register* on page 3-14
- *Configuration Code Register* on page 3-17
- *ASIC Control Register* on page 3-18
- *ETM ID Register* on page 3-19
- *Configuration Code Extension Register* on page 3-20
- *Extended External Input Selection Register* on page 3-22
- *Power-Down Status Register* on page 3-22
- *Auxiliary Control Register* on page 3-23
- *ETM ID Register 2* on page 3-24
- *Peripheral Identification Registers* on page 3-24
- *Component Identification Registers* on page 3-26
- *Integration Test Registers* on page 3-27.

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-A5 processor supports parallel instruction execution. This means the processor is capable of tracing two instructions per cycle.

Although the trace start/stop block is evaluated for each instruction as required, the ETM-A5 macrocell cannot trace one instruction without the other. In other words, if one instruction is traced, the instruction it is paired with is always traced as well. 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 v3.5, 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-A5 macrocell:

- *Trace features*
- *Comparator features*.

Trace features

ETM-A5 implements all of the ETMv3.5 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-A5 implements data address comparison on the CoreSight ETM-A5 macrocell. For a description of data address comparison see the *Embedded Trace Macrocell Architecture Specification*.

2.4.6 Interaction with the *Performance Monitoring Unit* (PMU)

The Cortex-A5 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[29:0]** input is mapped to the corresponding extended external input. See the *Cortex-A5 Technical Reference Manual* for details of the mapping of events to bits within this bus.

The Cortex-A5 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-A5 PMU.

2.4.7 Jazelle tracing

Data is not traced when a `BXJ` instruction is executed.

2.4.8 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*:

- 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 for which the data address is not sequential to the previously traced data transfer.
- When a branch packet is traced because of entry to a prohibited region, the packet always indicates an ISA of ARM state.
- The cycle counter continues to count while non-invasive debug is disabled.
- The **ETMDBGRQ** signal is deasserted when the programming bit is set.

2.5 Constraints and limitations of use

This section describes the constraints and limitations of use that apply to the ETM-A5.

2.5.1 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 [17:16, 13], take the value b000, indicating dynamic port mode.

For more information see *ETM Main Control Register* on page 3-14.

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.

3.2 Mode of operation

The following sections describe how you control ETM programming.

- *Controlling ETM programming*
- *Programming and reading ETM registers*.

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 has been 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 3-14.

The processor does not have to be in 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.

Note

Access to the ETM registers using coprocessor instructions, or the Direct JTAG access method, is not supported.

3.3 Data structures

See the *Embedded Trace Macrocell Architecture Specification* for descriptions of the trace packet formats generated by the ETM-A5 macrocell.

3.4 Register summary

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

- *Register descriptions* on page 3-14, for the implementation-defined registers
- the *Embedded Trace Macrocell Architecture Specification*, for the other registers.

Table 3-1 shows all of the registers, and tells you where each register is described in detail. 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 3-10. The functional group register tables include additional information about each register:

- The register number.
- The register access type, which is read-only, write-only, or read/write.
- 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:
  - 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.
  - 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-A5 register summary

Number	Name	Type	Reset	Group[a]	Description
0x000	ETMCR	RW	0x00000441	1	*ETM Main Control Register* on page 3-14
0x001	ETMCCR	RO	0x8D014024[b]	1	*Configuration Code Register* on page 3-17
0x002	ETMTRIGGER	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x003	ETMASICCTLR	RW	-[c]	1	*ASIC Control Register* on page 3-18
0x004	ETMSR	RW	-[c]	1	*Embedded Trace Macrocell Architecture Specification*
0x005	ETMSCR	RO	0x00020C0C[d]	1	*Embedded Trace Macrocell Architecture Specification*
0x006	ETMTSSCR	RW	-[c]	2	*Embedded Trace Macrocell Architecture Specification*
0x007	ETMTECR2	RW	-[c]	2	*Embedded Trace Macrocell Architecture Specification*
0x008	ETMTEEVR	RW	-[c]	2	*Embedded Trace Macrocell Architecture Specification*
0x009	ETMTECR1	RW	-[c]	2	*Embedded Trace Macrocell Architecture Specification*
0x00B	ETMFFLR[e]	RW	-[c]	1	*Embedded Trace Macrocell Architecture Specification*

Table 3-1 ETM-A5 register summary (continued)

Number	Name	Type	Reset	Group[a]	Description
0x00C	ETMVDEVR	RW	-[c]	2	*Embedded Trace Macrocell Architecture Specification*
0x00D	ETMVDCR1	RW	-[c]	2	*Embedded Trace Macrocell Architecture Specification*
0x00F	ETMVDCR3	RW	-[c]	2	*Embedded Trace Macrocell Architecture Specification*
0x010 to 0x017	ETMACVR1-8	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x020 to 0x027	ETMACTR1-8	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x030[f]	ETMDCVR1[f]	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x032[f]	ETMDCVR3[f]	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x040[f]	ETMDCMR1[f]	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x042[f]	ETMDCMR3[f]	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x050, 0x051	ETMCNTRLDVR1-2	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x054, 0x055	ETMCNTENR1-2	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x058, 0x059	ETMCNTRLDEVR1-2	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x05C, 0x05D	ETMCNTVR1-2	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x060 to 0x065	ETMSQabEVR	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x067	ETMSQR	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x068, 0x069	ETMEXTOUTEVR1-2	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x06C	ETMCIDCVR	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x06F	ETMCIDCMR	RW	-[c]	3	*Embedded Trace Macrocell Architecture Specification*
0x078	ETMSYNCFR	RW	0x00000400	1	*Embedded Trace Macrocell Architecture Specification*
0x079	ETMIDR	RO	0x410CF25x[g]	1	*ETM ID Register* on page 3-19
0x07A	ETMCCER	RO	0x304008F2	1	*Configuration Code Extension Register* on page 3-20
0x07B	ETMEXTINSELR	RW	-[c]	4	*Extended External Input Selection Register* on page 3-22
0x07E	ETMTSEVR	RW	-[c]	4	*Embedded Trace Macrocell Architecture Specification*
0x07F	ETMAUXCR	RW	-	1	*Auxiliary Control Register* on page 3-23
0x080	ETMTRACEIDR	RW	0x00000000	1	*Embedded Trace Macrocell Architecture Specification*
0x082	ETMIDR2	RO	-	1	*ETM ID Register 2* on page 3-24
0x0C5	ETMPDSR	RO	-[c]	1	*Power-Down Status Register* on page 3-22

Table 3-1 ETM-A5 register summary (continued)

Number	Name	Type	Reset	Group[a]	Description
0x3B7	ITMISCOUT	WO	n/a[j]	6	*ITMISCOUT Register, miscellaneous outputs* on page 3-29
0x3B8	ITMISCIN	RO[h]	-[i]	6	*ITMISCIN Register, miscellaneous inputs* on page 3-30
0x3BA	ITTRIGGERREQ	WO	n/a[j]	6	*ITTRIGGERREQ Register, trigger request* on page 3-30
0x3BB	ITATBDATA0	WO	n/a[j]	6	*ITATBDATA0 Register, ATB data 0* on page 3-32
0x3BC	ITATBCTR2	RO[h]	-[i]	6	*ITATBCTR2 Register, ATB control 2* on page 3-32
0x3BD	ITATBCTR1	WO	n/a[j]	6	*ITATBCTR1 Register, ATB control 1* on page 3-33
0x3BE	ITATBCTR0	WO	n/a[j]	6	*ITATBCTR0 Register, ATB control 0* on page 3-34
0x3C0	ETMITCTRL	RW	0x00000000	5	*Embedded Trace Macrocell Architecture Specification*
0x3E8	ETMCLAIMSET	RW	0x000000FF	5	*Embedded Trace Macrocell Architecture Specification*
0x3E9	ETMCLAIMCLR	RW	0x00000000	5	*Embedded Trace Macrocell Architecture Specification*
0x3EC	ETMLAR	WO	n/a[j]	5	*Embedded Trace Macrocell Architecture Specification*
0x3ED	ETMLSR	RO	-[i]	5	*Embedded Trace Macrocell Architecture Specification*
0x3EE	ETMAUTHSTATUS	RO	-[i]	5	*Embedded Trace Macrocell Architecture Specification*
0x3F2	ETMDEVID	RO	0x00000000	5	*Embedded Trace Macrocell Architecture Specification*
0x3F3	ETMDEVTYPE	RO	0x00000013	5	*Embedded Trace Macrocell Architecture Specification*

Table 3-1 ETM-A5 register summary (continued)

Number	Name	Type	Reset	Group[a]	Description
0x3F4 to 0x3F7	ETMPIDR4-7	RO	-[i]	5	*Peripheral Identification Registers* on page 3-24
0x3F8 to 0x3FB	ETMPIDR0-3	RO	-[i]		
0x3FC to 0x3FF	ETMCIDR0-3	RO	-[i]	5	*Component Identification Registers* on page 3-26

a. Functional group. For more information, see:
for Group 1, *General control and ID registers* on page 3-10
for Group 2, *TraceEnable and ViewData registers* on page 3-11
for Group 3, *Comparator registers* on page 3-11
for Group 4, *Counter, sequencer and other resource registers* on page 3-12
for Group 5, *CoreSight management registers* on page 3-13
for Group 6, *Integration test registers* on page 3-13, Table 3-8 on page 3-13.
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 *Signal descriptions* on page A-2
e. Although the macrocell does not include FIFOFULL logic, the FIFOFULL Level Register controls the FIFO level at which 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-A5, reserved areas are:
Register 0x031, Data Comparator Value 2, at offset 0x0C4 Register 0x033, Data Comparator Value 4, at offset 0x0CC
Register 0x041, Data Comparator Mask 2, at offset 0x104 Register 0x043, Data Comparator Mask 4, 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 3-19 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 details.
j. Not applicable. These are write-only registers.

Table 3-2 lists the descriptive names of the registers that are described in the *Embedded Trace Macrocell Architecture Specification*.

Table 3-2 Descriptive names of registers described in the *Embedded Trace Macrocell Architecture Specification*

Register number	Name	Description
0x002	ETMTRIGGER	Trigger Event Register
0x004	ETMSR	ETM Status Register
0x005	ETMSCR	System Configuration Register
0x006	ETMTSSCR	TraceEnable Start/Stop Control Register
0x007	ETMTECR2	TraceEnable Control 2 Register
0x008	ETMTEEVR	TraceEnable Event Register
0x009	ETMTECR1	TraceEnable Control 1 Register
0x00B	ETMFFLR	FIFOFULL Level Register
0x00C	ETMVDEVR	ViewData Event Register
0x00D	ETMVDCR1	ViewData Control 1 Register

Table 3-2 Descriptive names of registers described in the *Embedded Trace Macrocell Architecture Specification* (continued)

Register number	Name	Description
0x00F	ETMVDCR3	ViewData Control 3 Register
0x010 to 0x017	ETMACVR1-8	Address Comparator Value Registers 1-8
0x020 to 0x027	ETMACTR1-8	Address Comparator Access Type Registers 1-8
0x030[f]	ETMDCVR1	Data Comparator Value Register 1
0x032[f]	ETMDCVR3	Data Comparator Value Register 3
0x040	ETMDCMR1	Data Comparator Mask Register 1
0x042	ETMDCMR	Data Comparator Mask Register 3
0x050, 0x051	ETMCNTRLDVR1-2	Counter Reload Value Registers 1-2
0x054, 0x055	ETMCNTENR1-2	Counter Enable Registers 1-2
0x058, 0x059	ETMCNTRLDEVR1-2	Counter Reload Event Registers 1-2
0x05C, 0x05D	ETMCNTVR1-2	Counter Value Registers 1-2
0x060 to 0x065	ETMSQabEVR	Sequencer State Transition Event Registers
0x067	ETMSQR	Current Sequencer State Register
0x068, 0x069	ETMEXTOUTEVR1-2	External Output Event Registers 1-2
0x06C	ETMCIDCVR	Context ID Comparator Value Register
0x06F	ETMCIDCMR	Context ID Comparator Mask Register
0x078	ETMSYNCFR	Synchronization Frequency Register
0x080	ETMTRACEIDR	CoreSight Trace ID Register
0x3C0	ETMITCTRL	Integration Mode Control Register
0x3E8	ETMCLAIMSET	Claim Tag Set Register
0x3E9	ETMCLAIMCLR	Claim Tag Clear Register
0x3EC	ETMLAR	Lock Access Register
0x3ED	ETMLSR	Lock Status Register
0x3EE	ETMAUTHSTATUS	Authentication Status Register
0x3F2	ETMDEVID	CoreSight Device Configuration Register
0x3F3	ETMDEVTYPE	CoreSight Device Type Register
0x3F4 to 0x3F7	ETMPIDR4 to ETMPIDR7	*Peripheral Identification Registers* on page 3-24
0x3F8 to 0x3FB	ETMPIDR0 to ETMPIDR3	
0x3FC to 0x3FF	ETMCIDR0 to ETMCIDR3	*Component Identification Registers* on page 3-26

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 3-11
- *Comparator registers* on page 3-11
- *Counter, sequencer and other resource registers* on page 3-12
- *CoreSight management registers* on page 3-13
- *Integration test registers* on page 3-13.

These functional groups include all of the registers.

Note

All registers are in the **CLK** clock domain.

General control and ID registers

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

Table 3-3 General control and ID registers

Register number	Name	Base offset	Description
0x000	ETM Control	0x000	*ETM Main Control Register* on page 3-14
0x001	Configuration Code	0x004	*Configuration Code Register* on page 3-17
0x003	ASIC Control	0x00C	*ASIC Control Register* on page 3-18
0x004	ETM Status	0x010	*Embedded Trace Macrocell Architecture Specification*
0x005	System Configuration	0x014	*Embedded Trace Macrocell Architecture Specification*
0x00B	FIFOFULL Level[a]	0x02C	*Embedded Trace Macrocell Architecture Specification*
0x078	Synchronization Frequency	0x1E0	*Embedded Trace Macrocell Architecture Specification*
0x079	ETM ID	0x1E4	*ETM ID Register* on page 3-19
0x07A	Configuration Code Extension	0x1E8	*Configuration Code Extension Register* on page 3-20
0x07F	Auxiliary Control	0x1FC	*Auxiliary Control Register* on page 3-23
0x080	CoreSight Trace ID	0x200	*ETM Architecture Specification*
0x082	CoreSight Trace ID	0x208	*ETM ID Register 2* on page 3-24
0x0C5	Power-Down Status	0x314	*Power-Down Status Register* on page 3-22

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

TraceEnable and ViewData registers

Table 3-4 shows the TraceEnable and ViewData registers in register number order.

Table 3-4 TraceEnable and ViewData registers

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

Comparator registers

Table 3-5 shows the comparator registers in register number order. These control the Address, Data, and Context ID comparators.

Table 3-5 Comparator registers

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

a. In the Data Comparator register area, even number registers are reserved. For the CoreSight ETM-A5, reserved areas are:
Register 0x031, Data Comparator Value 2, at offset 0x0C4 Register 0x033, Data Comparator Value 4, at offset 0x0CC
Register 0x041, Data Comparator Mask 2, at offset 0x104 Register 0x043, Data Comparator Mask 4, 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 3-6 shows 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 3-6 Counter, sequencer and other resource registers

Register number	Name	Base offset	Description
0x002	Trigger Event	0x008	*Embedded Trace Macrocell Architecture Specification*
0x050, 0x051	Counter Reload Value 1-2	0x140, 0x144	*Embedded Trace Macrocell Architecture Specification*
0x054, 0x055	Counter Enable Event 1-2	0x150, 0x154	*Embedded Trace Macrocell Architecture Specification*
0x058, 0x059	Counter Reload Event 1-2	0x160, 0x164	*Embedded Trace Macrocell Architecture Specification*
0x05C, 0x05D	Counter Value 1-2	0x170, 0x174	*Embedded Trace Macrocell Architecture Specification*
0x060 to 0x065	Sequencer State Transition Events	0x180 to 0x194	*Embedded Trace Macrocell Architecture Specification*
0x067	Current Sequencer State	0x19C	*Embedded Trace Macrocell Architecture Specification*
0x068, 0x069	External Output Event 1-2	0x1A0, 0x1A4	*Embedded Trace Macrocell Architecture Specification*
0x07B	Extended External Input Selector	0x1EC	*Extended External Input Selection Register* on page 3-22
0x07E	Timestamp Event	0x1F8	*Embedded Trace Macrocell Architecture Specification*

CoreSight management registers

Table 3-7 shows the CoreSight management registers in register number order.

Table 3-7 CoreSight management registers

Register number	Name	Base offset	Description
0x3C0	Integration Mode Control	0xF00	*Embedded Trace Macrocell Architecture Specification*
0x3E8	Claim Tag Set	0xFA0	*Embedded Trace Macrocell Architecture Specification*
0x3E9	Claim Tag Clear	0xFA4	*Embedded Trace Macrocell Architecture Specification*
0x3EC	Lock Access	0xFB0	*Embedded Trace Macrocell Architecture Specification*
0x3ED	Lock Status	0xFB4	*Embedded Trace Macrocell Architecture Specification*
0x3EE	Authentication Status	0xFB8	*Embedded Trace Macrocell Architecture Specification*
0x3F2	Device Configuration	0xFC8	*Embedded Trace Macrocell Architecture Specification*
0x3F3	Device Type	0xFCC	*Embedded Trace Macrocell Architecture Specification*
0x3F4 to 0x3F7	Peripheral ID4 to 7	0xFD0 to 0xFDC	*Peripheral Identification Registers* on page 3-24
0x3F8 to 0x3FB	Peripheral ID0 to 3	0xFE0 to 0xFEC	
0x3FC to 0x3FF	Component ID0 to 3	0xFF0 to 0xFFC	*Component Identification Registers* on page 3-26

Integration test registers

Table 3-8 shows the integration test registers in register number order.

Table 3-8 Integration test registers

Register number	Name	Base offset	Description
0x3B7	ITMISCOUT	0xEDC	*ITMISCOUT Register, miscellaneous outputs* on page 3-29
0x3B8	ITMISCIN	0xEE0	*ITMISCIN Register, miscellaneous inputs* on page 3-30
0x3BA	ITTRIGGERREQ	0xEE8	*ITTRIGGERREQ Register, trigger request* on page 3-30
0x3BB	ITATBDATA0	0xEEC	*ITATBDATA0 Register, ATB data 0* on page 3-32
0x3BC	ITATBCTR2	0xEF0	*ITATBCTR2 Register, ATB control 2* on page 3-32
0x3BD	ITATBCTR1	0xEF4	*ITATBCTR1 Register, ATB control 1* on page 3-33
0x3BE	ITATBCTR0	0xEF8	*ITATBCTR0 Register, ATB control 0* on page 3-34

3.5 Register descriptions

The following sections describe the implementation-defined ETM-A5 registers:

- *ETM Main Control Register*
- *Configuration Code Register* on page 3-17
- *ASIC Control Register* on page 3-18
- *ETM ID Register* on page 3-19
- *Configuration Code Extension Register* on page 3-20
- *Extended External Input Selection Register* on page 3-22
- *Power-Down Status Register* on page 3-22
- *Auxiliary Control Register* on page 3-23
- *ETM ID Register 2* on page 3-24
- *Peripheral Identification Registers* on page 3-24
- *Component Identification Registers* on page 3-26
- *Integration Test Registers* on page 3-27.

The *Embedded Trace Macrocell Architecture Specification* describes the other ETM-A5 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 3-5 and Table 3-3 on page 3-10.

Figure 3-1 shows the ETMCR bit assignments.

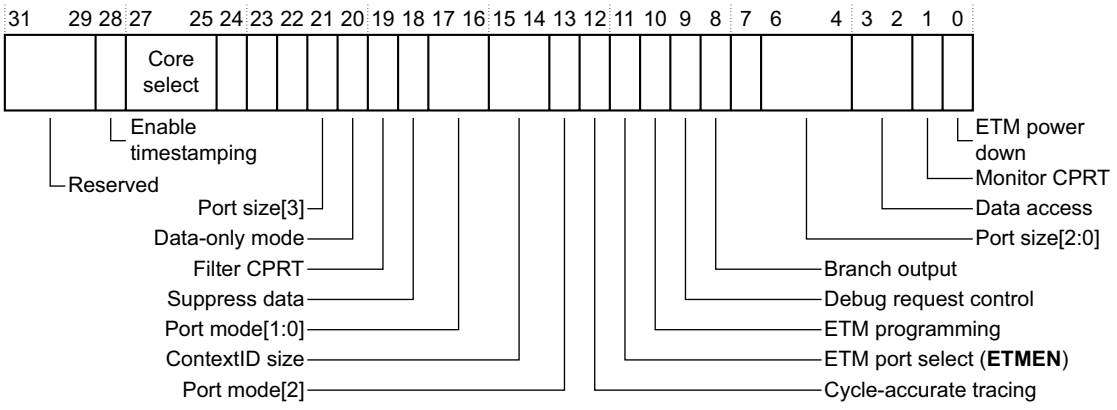


Figure 3-1 ETMCR bit assignments

Table 3-9 shows the ETMCR bit assignments.

Table 3-9 ETMCR bit assignments

Bits	Name	Access	Description
[31:29]	Reserved	RW	Must be written as 0.
[28]	Enable timestamping	RW	Set to 1 to enable timestamping. On an ETM reset this bit is 0.
[27:25]	Core select	RW	If an ETM is shared between multiple cores, selects which core to trace. For the maximum value permitted, see bits [14:12] of the System Configuration Register. See the *Embedded Trace Macrocell Architecture Specification* for more information. To guarantee that the ETM is correctly synchronized to the new core, you must update these bits as follows: 1. Set bit [10], ETM programming, and bit [0], ETM power down, to 1. 2. Change the core select bits. 3. Clear bit [0], ETM power down, to 0. 4. Perform other programming required as normal. On an ETM reset this field is zero.
[24]	Instrumentation resources access control	RO	ETM-A5 does not implement any instrumentation resources and therefore this bit is RAZ.
[23]	Disable software writes	RO	ETM-A5 does not support this feature and therefore this bit is RAZ.
[22]	Disable register writes from the debugger	RO	ETM-A5 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. For details see *Filter Coprocessor Register Transfers* (CPRT) in ETMv3.0 and later in the *Embedded Trace Macrocell Architecture Specification*. On an ETM reset this bit is 0.
[18]	Suppress data	RW	Use this bit with bit [7] to suppress data. For details see Data suppression in the *Embedded Trace Macrocell Architecture Specification*. 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-A5 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 3-9 ETMCR bit assignments (continued)

Bits	Name	Access	Description
[15:14]	ContextIDsize	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 is 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 Cortex-A5 processor to be forced into Debug state. On an ETM reset this bit is 0.
[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-A5 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-A5 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 other values 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.

Table 3-9 ETMCR bit assignments (continued)

Bits	Name	Access	Description
[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]. For details see *Filter Coprocessor Register Transfers* (CPRT) in ETMv3.0 and later in the *Embedded Trace Macrocell Architecture Specification*. On an ETM reset this bit is 0.
[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.

Usage constraints There are no usage constraints.

Configurations Always available.

Attributes See the register summary in Table 3-1 on page 3-5 and Table 3-3 on page 3-10.

If the **MAXEXTOUT[1:0]** and **MAXEXTIN[2:0]** signals are all tied LOW (0) the ETMCCR has the value `0x8D014024`.

Figure 3-2 on page 3-18 shows the ETMCCR bit assignments.

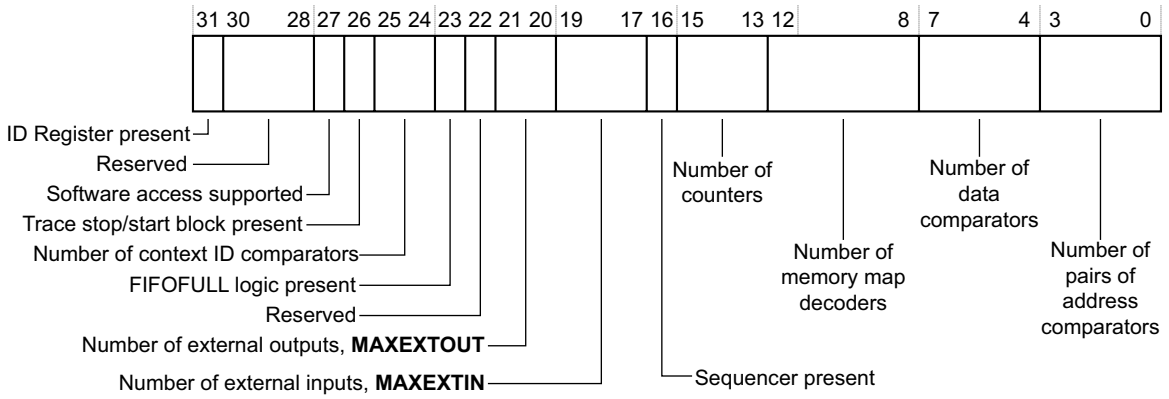


Figure 3-2 ETMCCR bit assignments

Table 3-10 shows the ETMCCR bit assignments.

Table 3-10 ETMCCR bit assignments

Bits	Value	Description
[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 value of these bits is the minimum of **MAXEXTOUT[1:0]** and 2, because ETM-A5 supports a maximum of 2 external outputs.
[19:17]	-	Number of external inputs. Determined by the **MAXEXTIN[2:0]** inputs. The value of these bits is the minimum of **MAXEXTIN[2:0]** and 4, because ETM-A5 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 3-5 and Table 3-3 on page 3-10.

Figure 3-3 shows the ETMASICCR bit assignments.

Figure 3-3 ETMASICCR bit assignments

Table 3-11 shows the ETMASICCR bit assignments.

Table 3-11 ETMASICCR bit assignments

Bits	Description
[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.

3.5.4 ETM ID Register

The ETMIDR characteristics are:

Purpose Identifies the implementation of the ETM.

Usage constraints There are no usage constraints.

Configurations Always available.

Attributes This register has the value `0x410CF25x`, where x depends on the release version of the macrocell, see the Implementation revision field description in Table 3-12 on page 3-20 for more information.

See the register summary in Table 3-1 on page 3-5 and Table 3-3 on page 3-10.

Figure 3-4 shows the ETMIDR bit assignments.

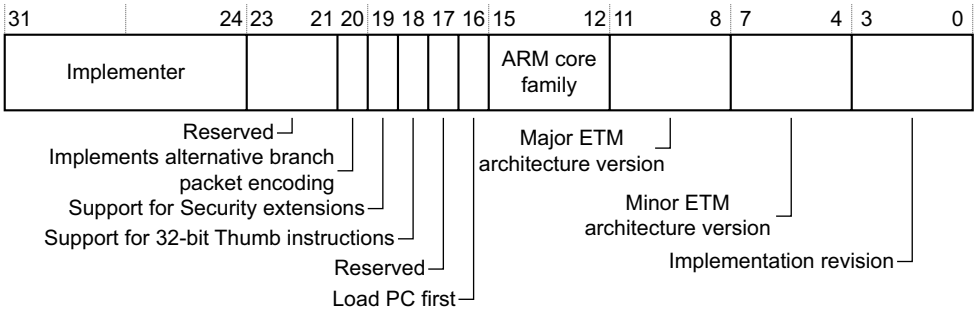


Figure 3-4 ETMIDR bit assignments

Table 3-12 shows the ETMIDR bit assignments.

Table 3-12 ETMIDR bit assignments

Bits	Value	Description
[31:24]	0x41	Implementer = A (for ARM).
[23:21]	b0000	Reserved.
[20]	0	Branch packet encoding implemented. This bit is set to 0, indicating that ETM-A5 implements the original branch packet encoding.
[19]	1	Support for Security Extensions. This bit is set to 1, indicating that the ARM architecture Security Extensions are implemented by the processor.
[18]	1	Support for 32-bit Thumb instructions. 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	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 ETM v3.x, this field = 2.
[7:4]	b0101	Minor ETM architecture version number. For ETM vx.5, this field = 5.
[3:0]	b0010	Implementation revision. Value given is for the r0p2 release of the macrocell.

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 3-5 and Table 3-3 on page 3-10.

Figure 3-5 on page 3-21 shows the ETMCCER bit assignments.

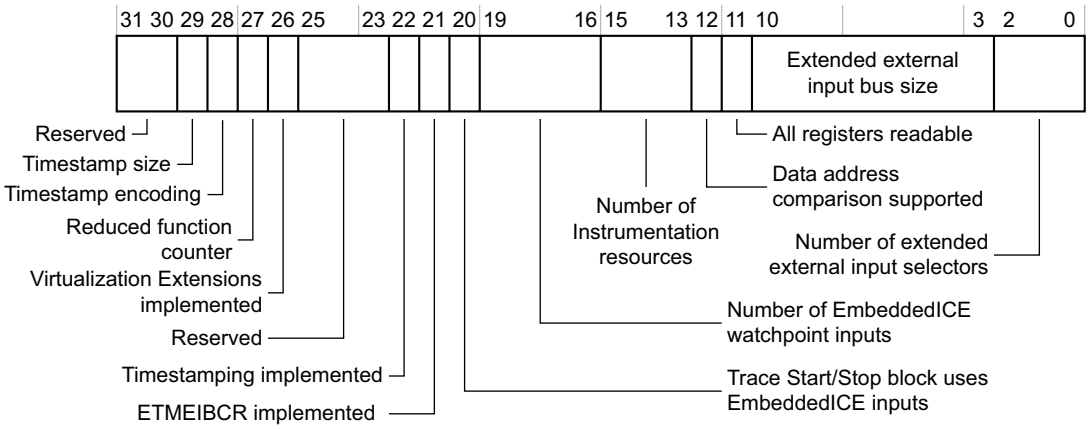


Figure 3-5 ETMCCER bit assignments

Table 3-13 shows the ETMCCER bit assignments.

Table 3-13 ETMCCER bit assignments

Bits	Value	Description
[31:30]	0	Reserved, RAZ.
[29]	-	Timestamp size. The possible values of this bit are: **0** Value is 48-bits. **1** Value is 64-bits.
[28]	-	Timestamp encoding. The possible values of this bit are: **0** The value is gray encoded. **1** The value is natural binary encoded.
[27]	0	Reduced function counter. This bit is 0, indicating that all counters are implemented as full-function counters.
[26]	0	The Virtualization Extensions are implemented. This bit is 0, indicating that the Virtualization Extensions are not implemented.
[25:23]	0	Reserved, RAZ.
[22]	1	Timestamping implemented. In the ETM-A5, this bit is always set to 1, and: • the Timestamp Event Register is implemented • bit [28] in the Main Control Register is writable.
[21]	0	ETMEIBCR implemented. This bit is 1, indicating that the register is not implemented.
[20]	0	Trace Start/Stop block uses EmbeddedICE watchpoint inputs. This bit is 0, indicating that the Trace Start/Stop block cannot use these inputs.
[19:16]	0	Number of EmbeddedICE watchpoint inputs. This field is b0000, indicating that no EmbeddedICE watchpoint inputs are supported..
[15:13]	0	Number of Instrumentation resources supported. This field is b000, indicating that no instrumentation resources are supported..
[12]	0	Data address comparisons supported. This bit is 0, indicating that data address comparisons are supported.

Table 3-13 ETMCCER bit assignments (continued)

Bits	Value	Description
[11]	1	All registers, except some integration test registers, are readable. See Table 3-8 on page 3-13 for details of the access to integration test registers[a].
[10:3]	30	Size of extended external input bus.
[2:0]	2	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 Register 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 3-5 and Table 3-6 on page 3-12.

Figure 3-6 shows the ETMEXTINSELR bit assignments.

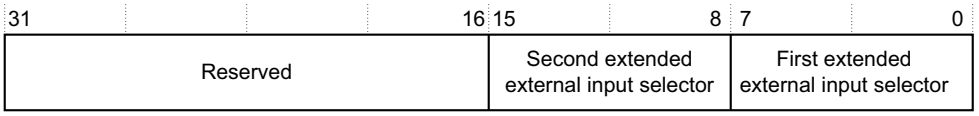


Figure 3-6 ETMEXTINSELR bit assignments

Table 3-14 shows the ETMEXTINSELR bit assignments.

Table 3-14 ETMEXTINSELR bit assignments

Bits	Description
[31:16]	Reserved, SBZP.
[15:8]	Second extended external input selector: **Bits [15:13]** Reserved, SBZP. **Bits [12:8]** Selection value for second external input.
[7:0]	First extended external input selector: **Bits [7:5]** Reserved, SBZP. **Bits [4:0]** 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 3-5 and Table 3-3 on page 3-10.

Figure 3-7 shows the ETMPDSR bit assignments.

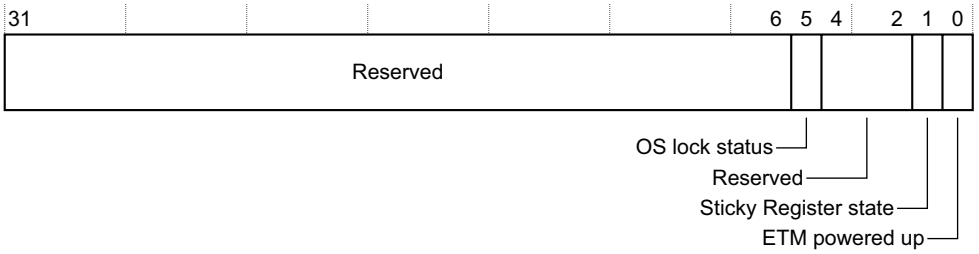


Figure 3-7 ETMPDSR bit assignments

Table 3-15 shows the ETMPDSR bit assignments.

Table 3-15 ETMPDSR bit assignments

Bits	Value	Description
[31:6]	0	Reserved, RAZ
[5]	LK	OS lock status. ETM-A5 does not implement the OS lock mechanism, so this bit is RAZ.
[4:2]	0	Reserved, RAZ
[1]	0	Sticky Register State. ETM-A5 does not support multiple power domains so this bit is RAZ.
[0]	1	ETM powered up. The ETM Trace registers are accessible. ETM-A5 does not support multiple power domains so this bit is RAO.

3.5.8 Auxiliary Control Register

The ETMAUXCR characteristics are:

Purpose Provides additional implementation-defined ETM controls.

Usage constraints There are no usage constraints.

Configurations Always available.

Attributes See the register summary in Table 3-1 on page 3-5 and Table 3-3 on page 3-10.

Figure 3-8 shows the ETMAUXCR bit assignments.

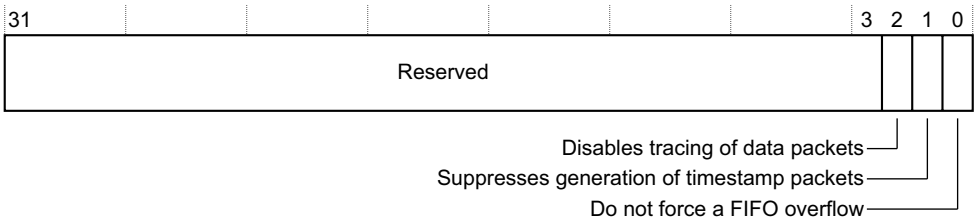


Figure 3-8 ETMAUXCR bit assignments

Table 3-16 shows the ETMAUXCR bit assignments.

Table 3-16 ETMAUXCR bit assignments

Bits	Value	Description
[31:3]	-	Reserved, RAZ.
[2]	0	Disables tracing of data packets from VFP or Neon load or store instructions. This only suppresses the data packets. Comparators and other resources can still match on the data from these instructions.
[1]	0	Suppresses generation of timestamp packets because of the execution of `ISB` instructions.
[0]	0	Do not force a FIFO overflow if periodic synchronization has not been output for two full synchronization periods.

3.5.9 ETM ID Register 2

The ETMIDR2 characteristics are:

Purpose Provides an extension to the ETM ID register, ETMIDR.

Usage constraints There are no usage constraints.

Configurations Always available.

Attributes See the register summary in Table 3-1 on page 3-5 and Table 3-3 on page 3-10.

Figure 3-9 shows the ETMIDR2 bit assignments.

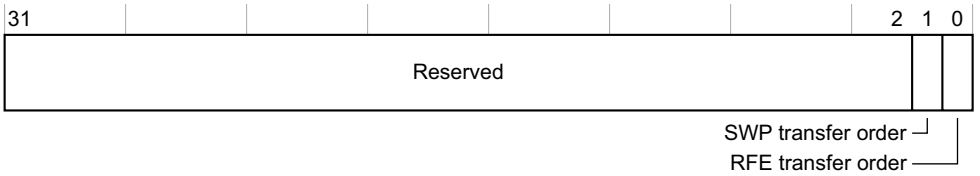


Figure 3-9 ETMIDR2 bit assignments

Table 3-17 shows the ETMIDR2 bit assignments.

Table 3-17 ETMIDR2 bit assignments

Bits	Value	Description
[31:2]	-	Reserved, RAZ.
[1]	1	Identifies the order of transfers for a `SWP` or `SWPB` instruction: This bit is 1, indicating that the Store transfer is traced before the Load transfer.
[0]	1	Identifies the order of transfers for the `RFE` instruction: This bit is 1, indicating that the CPSR transfer is traced before the PC transfer.

3.5.10 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 3-10 shows.

Configurations Always available.

Attributes See the register summary in Table 3-1 on page 3-5 and Table 3-7 on page 3-13.

Figure 3-10 shows the mapping between ETMPIDR0-ETMPIDR7 and the single 64-bit *Peripheral ID* value,

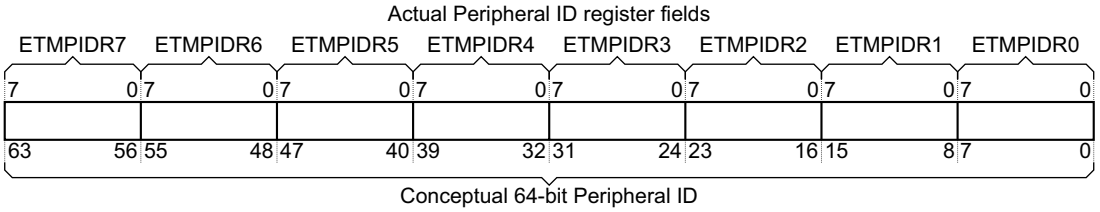


Figure 3-10 Mapping between ETMPIDR0-ETMPIDR7 and the Peripheral ID value

Figure 3-11 shows the Peripheral ID bit assignments in the single conceptual Peripheral ID register.

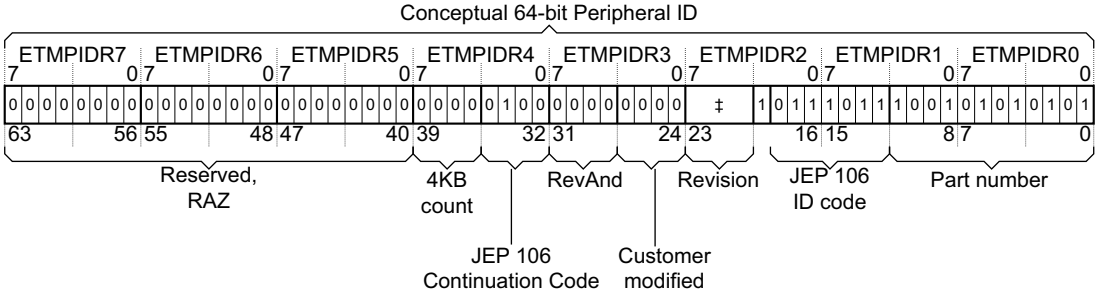


Figure 3-11 Peripheral ID fields

Table 3-18 shows 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 3-18 ETMPIDR0-ETMPIDR7 bit assignments

Register	Register number	Register offset	Bits	Value	Description
ETMPIDR7	0x3F7	0xFDC	[31:8]	-	Reserved.
			[7:0]	0x00	Reserved.
ETMPIDR6	0x3F6	0xFD8	[31:8]	-	Reserved.
			[7:0]	0x00	Reserved.
ETMPIDR5	0x3F5	0xFD4	[31:8]	-	Reserved.
			[7:0]	0x00	Reserved.

Table 3-18 ETMPIDR0-ETMPIDR7 bit assignments (continued)

Register	Register number	Register offset	Bits	Value	Description
ETMPIDR4	0x3F4	0xFD0	[31:8]	-	Reserved.
			[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]	-	Reserved.
			[7:4]	0x0	RevAnd (at top level). Manufacturer revision number.
			[3:0]	0x0	Customer Modified. 0x0 indicates from ARM.
ETMPIDR2	0x3FA	0xFE8	[31:8]	-	Reserved.
			[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 3-19.
			[3]	1	Always 1. Indicates that a JEDEC assigned value is used.
			[2:0]	b011	JEP 106 identity code[6:4].
ETMPIDR1	0x3F9	0xFE4	[31:8]	-	Reserved.
			[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]	-	Reserved.
			[7:0]	0x55	Part Number[7:0]. Middle and Lower BCD value of Device Number.

a. See the Description column for details.

Note

In Table 3-18 on page 3-25, the *Peripheral Identification Registers* on page 3-24 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 3-19 on page 3-27 the *Component Identification Registers* are listed in order of register name, from most significant (ETMCIDR3) to least significant (ETMCIDR0).

3.5.11 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 3-12 on page 3-27 shows.

Configurations Always available.

Attributes See the register summary in Table 3-1 on page 3-5 and Table 3-7 on page 3-13.

Figure 3-12 shows the mapping between ETMCIDR0-ETMCIDR3 and the single 32-bit *Component ID* value.

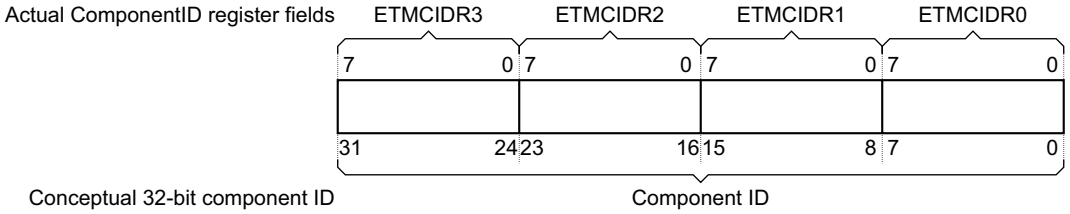


Figure 3-12 Mapping between ETMCIDR0-ETMCIDR3 and the Component ID value

Table 3-19 shows the Component ID bit assignments in the single conceptual Component ID register.

Table 3-19 ETMCIDR0-ETMCIDR3, bit assignments

Register	Register number	Register offset	Bits	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.
			[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].

3.5.12 Integration Test Registers

The following subsections describe the Integration Test Registers. To access these registers, 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 3-20 on page 3-28 shows 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 3-21 on page 3-28 shows the signals that can be read in this way.

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

Table 3-20 Output signals that the Integration Test Registers can control

Signal	Register	Bits	Register description
AFREADY	ITATBCTR0	[1]	See *ITATBCTR0 Register, ATB control 0* on page 3-34
ATBYTES[1:0]	ITATBCTR0	[9:8]	See *ITATBCTR0 Register, ATB control 0* on page 3-34
ATDATA[31, 23, 15, 7, 0]	ITATBDATA0	[4:0]	See *ITATBDATA0 Register, ATB data 0* on page 3-32
ATID[6:0]	ITATBCTR1	[6:0]	See *ITATBCTR1 Register, ATB control 1* on page 3-33
ATVALID	ITATBCTR0	[0]	See *ITATBCTR0 Register, ATB control 0* on page 3-34
ETMDBGRQ	ITMISCOUT	[4]	See *ITMISCOUT Register, miscellaneous outputs* on page 3-29
ETMSTANDBYWFX	ITMISCOUT	[5]	See *ITMISCOUT Register, miscellaneous outputs* on page 3-29
EXTOUT[1:0]	ITMISCOUT	[9:8]	See *ITMISCOUT Register, miscellaneous outputs* on page 3-29
SYNCREQ	ITATBCTR2	[2]	See *ITATBCTR2 Register, ATB control 2* on page 3-32
TRIGGER	ITTRIGGERREQ	[0]	See *ITTRIGGERREQ Register, trigger request* on page 3-30

Table 3-21 Input signals that the Integration Test Registers can read

Signal	Register	Bits	Register description
AFVALID	ITATBCTR2	[1]	See *ITATBCTR2 Register, ATB control 2* on page 3-32
ATREADY	ITATBCTR2	[0]	See *ITATBCTR2 Register, ATB control 2* on page 3-32
DBGACK	ITMISCIN	[4]	See *ITMISCIN Register, miscellaneous inputs* on page 3-30
ETMWFXPENDING	ITMISCIN	[5]	See *ITMISCIN Register, miscellaneous inputs* on page 3-30
EXTIN[3:0]	ITMISCIN	[3:0]	See *ITMISCIN Register, miscellaneous inputs* on page 3-30

Using the Integration Test Registers

The *CoreSight ETM-A5 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 details of ETMITCTRL.

ITMISCOUT Register, miscellaneous outputs

The ITMISCOUT characteristics are:

Purpose Sets the state of the output pins shown in Table 3-22.

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 3-5, Table 3-8 on page 3-13, and Table 3-21 on page 3-28.

Figure 3-13 shows the ITMISCOUT bit assignments.

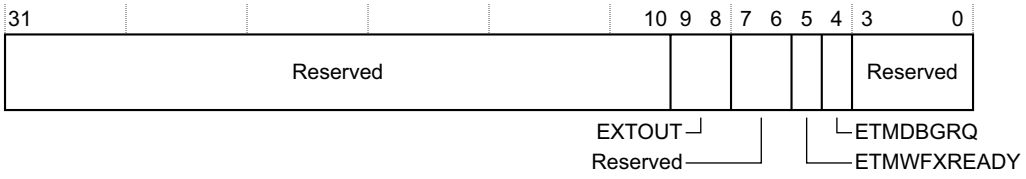


Figure 3-13 ITMISCOUT bit assignments

Table 3-22 shows the ITMISCOUT bit assignments.

Table 3-22 ITMISCOUT bit assignments

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

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 ITMISCOUT bit values correspond to the physical state of the output pins.

ITMISCIN Register, miscellaneous inputs

The ITMISCIN characteristics are:

Purpose Reads the state of the input pins shown in Table 3-23

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 3-5, Table 3-8 on page 3-13, and Table 3-21 on page 3-28.

Figure 3-14 shows the ITMISCIN bit assignments.

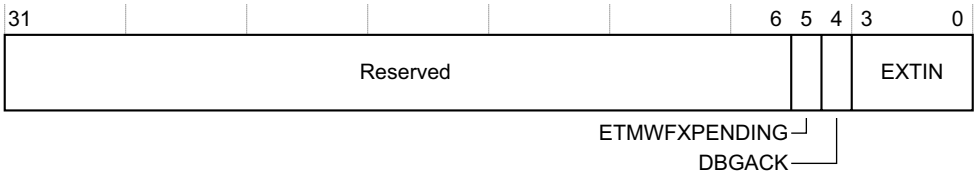


Figure 3-14 ITMISCIN bit assignments

Table 3-23 shows the ITMISCIN bit assignments.

Table 3-23 ITMISCIN bit assignments

Bits	Name	Description
[31:6]	-	Reserved. Read undefined.
[5]	ETMWFXPENDING	Returns the value of the **ETMWFXPENDING** 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 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 ITMISCIN bit values always correspond to the physical state of the input pins.

ITTRIGGERREQ Register, trigger request

The ITTRIGGERREQ characteristics are:

Purpose Sets the **TRIGGER** output pin shown in Table 3-24 on page 3-31.

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 3-5, Table 3-8 on page 3-13, and Table 3-21 on page 3-28.

Figure 3-15 on page 3-31 shows the ITTRIGGERREQ bit assignments.

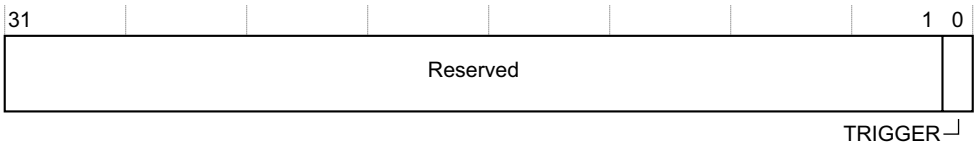


Figure 3-15 ITTRIGGERREQ bit assignments

Table 3-24 shows the ITTRIGGERREQ bit assignments.

Table 3-24 ITTRIGGERREQ bit assignments

Bits	Name	Description
[31:1]	-	Reserved. Write as zero.
[0]	TRIGGER	Drives the **TRIGGER** 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 ITTRIGGERREQ bit values always correspond to the physical state of the output pins.

ITATBDATA0 Register, ATB data 0

The ITATBDATA0 characteristics are:

Purpose Sets the state of the ATDATA output pins shown in Table 3-25.

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 3-5, Table 3-8 on page 3-13, and Table 3-21 on page 3-28.

Figure 3-16 shows the ITATBDATA0 bit assignments.

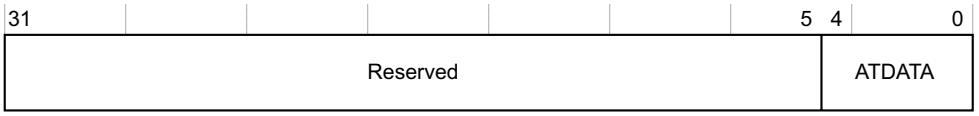


Figure 3-16 ITATBDATA0 bit assignments

Table 3-25 shows the ITATBDATA0 bit assignments.

Table 3-25 ITATBDATA0 bit assignments

Bits	Name	Description
[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.

ITATBCTR2 Register, ATB control 2

The ITATBCTR2 characteristics are:

Purpose Reads the state of the **AFVALID**, **ATREADY**, and **SYNCREQ** input pins from the ATB bus, as shown in Table 3-26 on page 3-33.

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 3-5, Table 3-8 on page 3-13, and Table 3-21 on page 3-28.

Figure 3-17 on page 3-33 shows the ITATBCTR2 bit assignments.

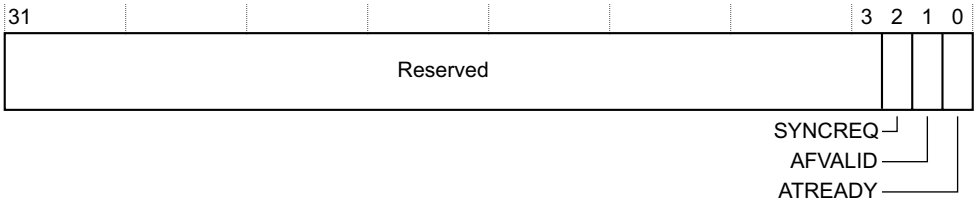


Figure 3-17 ITATBCTR2 bit assignments

Table 3-26 shows the ITATBCTR2 bit assignments.

Table 3-26 ITATBCTR2 bit assignments

Bits	Name	Description
[31:3]	-	Reserved. Read undefined.
[2]	SYNCREQ	Returns the value of the **SYNCREQ** input pin
[1]	AFVALID	Returns the value of the **AFVALID** input pin[a].
[0]	ATREADY	Returns the value of the **ATREADY** 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 ITATBCTR2 bit values always correspond to the physical state of the input pins.

ITATBCTR1 Register, ATB control 1

The ITATBCTR1 characteristics are:

Purpose Sets the state of the **ATID** 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 3-5, Table 3-8 on page 3-13, and Table 3-21 on page 3-28.

Figure 3-18 shows the ITATBCTR1 bit assignments.

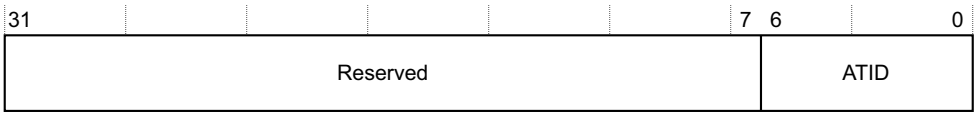


Figure 3-18 ITATBCTR1 bit assignments

Table 3-27 shows the ITATBCTR1 bit assignments.

Table 3-27 ITATBCTR1 bit assignments

Bits	Name	Description
[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.

ITATBCTR0 Register, ATB control 0

The ITATBCTR0 characteristics are:

Purpose Sets the state of the output pins shown in Table 3-28.

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 3-5, Table 3-8 on page 3-13, and Table 3-21 on page 3-28.

Figure 3-19 shows the ITATBCTR0 bit assignments.

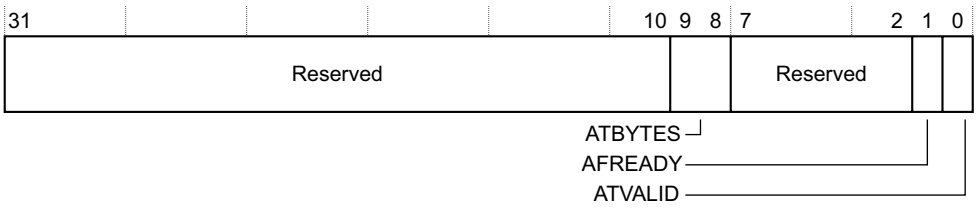


Figure 3-19 ITATBCTR0 bit assignments

Table 3-28 shows the ITATBCTR0 bit assignments.

Table 3-28 ITATBCTR0 bit assignments

Bits	Name	Description
[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 section:

- *Signal descriptions* on page A-2.

A.1 Signal descriptions

Table A-1 shows the ETM-A5 signals in alphabetical order. All input signals are sampled and output signals generated on one of three clocks **CLK**, **ATCLK**, or **PCLKDBG**. **ATCLK** and **PCLKDBG** must either be the same clock as **CLK**, or synchronously divided versions of **CLK**, using the **ATCLKEN** and **PCLKENDBG** enable signals. See the *CoreSight ETM-A5 Integration Manual* for information about signals and connectivity.

Table A-1 ETM-A5 signals

Signal	Type	Clock	Description
AFREADY	Output	**ATCLK**	ATB interface FIFO flush finished.
AFVALID	Input	**ATCLK**	ATB interface FIFO flush request.
ASICCTL[7:0]	Output	**CLK**	Contents of ASICCTL Register.
ATBYTES[1:0]	Output	**ATCLK**	Size of **ATDATA**.
ATCLKEN	Input	**CLK**	Clock enable for ATB interface.
ATDATA[31:0]	Output	**ATCLK**	ATB interface data.
ATID[6:0]	Output	**ATCLK**	ATB interface trace source ID.
ATREADY	Input	**ATCLK**	**ATDATA** can be accepted.
ATVALID	Output	**ATCLK**	ATB interface data valid.
CLK	Input	n/a	ETM clock. Same as Cortex-A5 processor clock.
CLKCHANGE	Input	**CLK**	Clock change indicator input. It is used when either the processor clock period or timestamp period changes.
CORESELECT[2:0]	Output	**CLK**	Where an ETM is shared between multiple cores, this signal specifies which core to trace. The value appears as bits [14:12] of the System Configuration Register.
DBGACK	Input	**CLK**	Indicates that the core is in debug state. This signal is connected to the core 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 core and the ETM.
DBGEN	Input	**CLK**	Invasive debug enable. When HIGH (1), indicates that invasive debug is enabled.
ETMCID[31:0]	Input	**CLK**	Current value of the processor Context ID Register.
ETMDA[31:0]	Input	**CLK**	Address for data transfer.
ETMDBGRQ	Output	**CLK**	Request from the macrocell for the core to enter debug state. This must be ORed with any ASIC-level **DBGRQ** signals before being connected to the core **EDBGRQ** input.
ETMDCTL[10:0]	Input	**CLK**	Data control signals.
ETMDD[31:0]	Input	**CLK**	Contains the data value for a Load, Store, `MRC`, or `MCR` instruction.
ETMEN	Output	**CLK**	Enable signal for trace output from the ETM, driven by bit [11] of the ETMCR.
ETMIA[31:0]	Input	**CLK**	Address for executed instruction.

Table A-1 ETM-A5 signals (continued)

Signal	Type	Clock	Description
ETMICTL[19:0]	Input	**CLK**	Instruction control signals.
ETMPWRUP	Output	**CLK**	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-A5 processor disables the interface • logic within the macrocell is clock-gated to conserve power.
ETMSTANDBYWFX	Output	**CLK**	Indicates that the macrocell FIFO is empty and that the Cortex-A5 processor can be put into Standby mode.
ETMWFXPENDING	Input	**CLK**	Indicates that the Cortex-A5 processor is about to go into Standby mode, and that the ETM must drain its FIFO.
EVNTBUS[29:0]	Input	**CLK**	Gives the status of the performance monitoring events. Used as extended external inputs.
EXTIN[3:0]	Input	**CLK**	External input resources.
EXTOUT[1:0]	Output	**CLK**	External outputs.
FIFOPEEK[6:0]	Output	**CLK**	For validation purposes only. Indicates when various events occur before being written to the FIFO.
MAXCORES[2:0]	Input	**CLK**	Where an ETM is shared between multiple cores, this signal specifies the number of cores the ETM can trace. It must be tied to the number of cores 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 3-5.
MAXEXTIN[2:0]	Input	**CLK**	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 3-17.
MAXEXTOUT[1:0]	Input	**CLK**	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 3-17.
NIDEN	Input	**CLK**	Non-invasive debug enable. When HIGH (1), indicates that non-invasive debug is enabled.
nSYSPORESET	Input	n/a	Power-on (main) reset.
PADDRDBG[11:2]	Input	**PCLKDBG**	Debug APB Address Bus.
PADDRDBG31	Input	**PCLKDBG**	Indicates an external debug request 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.
PCLKENDBG	Input	**CLK**	Debug APB clock enable.
PENABLEDBG	Input	**PCLKDBG**	The Debug APB interface is enabled for a transfer.
PRDATADBG[31:0]	Output	**PCLKDBG**	Debug APB read data.
PREADYDBG	Output	**PCLKDBG**	Used to extend Debug APB transfers.
PSELDBG	Input	**PCLKDBG**	Debug APB slave select signal.

Table A-1 ETM-A5 signals (continued)

Signal	Type	Clock	Description
PWDATADBG[31:0]	Input	**PCLKDBG**	Debug APB write data.
PWRITEDBG	Input	**PCLKDBG**	Debug APB transfer direction: 0 = Read 1 = Write.
DFTRSTDISABLE	Input	**CLK**	Reset synchronization bypass DFT signal.
DFTSE	Input	**CLK**	Scan enable DFT signal.
SYNCREQ	Input	**ATCLK**	Request for periodic synchronization.
TRIGGER	Output	**ATCLK**	Trigger request status signal. Asserted for one clock cycle when a trigger occurs..
TSMAXWIDTH	Input	**CLK**	Timestamp maximum size: 0 = 48 bits 1 = 64 bits.
TSNATURAL	Input	**CLK**	The timestamp is natural binary encoded: 0 = Gray encoded 1 = Binary encoded.
TSVALUE[63:0]	Input	**CLK**	Timestamp value input bus.

Appendix B
Revisions

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

Table B-1 Issue A

Change	Location	Affects
No changes, first release	-	-

Table B-2 Differences between issue A and issue B

Change	Location	Affects
Implementation revision value updated	Table 3-12 on page 3-20	r0p1

Table B-3 Differences between issue B and issue C

Change	Location	Affects
Implementation revision value updated	Table 3-12 on page 3-20	r0p2

Glossary

This glossary describes some of the terms used in ARM manuals. Where terms can have several meanings, the meaning presented here is intended.

Advanced Microcontroller Bus Architecture (AMBA)
A family of protocol specifications that describe a strategy for the interconnect. AMBA is the ARM open standard for on-chip buses. It is an on-chip bus specification that details a strategy for the interconnection and management of functional blocks that make up a *System-on-Chip* (SoC). It aids in the development of embedded processors with one or more CPUs or signal processors and multiple peripherals. AMBA complements a reusable design methodology by defining a common backbone for SoC modules.

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

AMBA *See* Advanced Microcontroller Bus Architecture.

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

APB *See* Advanced Peripheral Bus.

Application Specific Integrated Circuit (ASIC)
An integrated circuit that has been designed to perform a specific application function. It can be custom-built or mass-produced.

Architecture The organization of hardware and/or software that characterizes a processor and its attached components, and enables devices with similar characteristics to be grouped together when describing their behavior, for example, Harvard architecture, instruction set architecture, ARMv6 architecture.

ARM instruction A word that specifies an operation for an ARM processor to perform. ARM instructions must be word-aligned.

ARM state A processor that is executing ARM (32-bit) word-aligned instructions is operating in ARM state.

ASIC *See* Application Specific Integrated Circuit.

ATB *See* Advanced Trace Bus.

ATB bridge A synchronous ATB bridge provides a register slice to facilitate timing closure through the addition of a pipeline stage. It also provides a unidirectional link between two synchronous ATB domains.

An asynchronous ATB bridge provides a unidirectional link between two ATB domains with asynchronous clocks. It is intended to support connection of components with ATB ports residing in different clock domains.

Breakpoint A breakpoint is a mechanism provided by debuggers to identify an instruction at which program execution is to be halted. Breakpoints are inserted by the programmer to enable inspection of register contents, memory locations, variable values at fixed points in the program execution to test that the program is operating correctly. Breakpoints are removed after the program is successfully tested.

See also Watchpoint.

Byte An 8-bit data item.

Cold reset Also known as power-on reset. Starting the processor by turning power on. Turning power off and then back on again clears main memory and many internal settings. Some program failures can lock up the processor and require a cold reset to enable the system to be used again. In other cases, only a warm reset is required.

See also Warm reset.

Communications channel The hardware used for communicating between the software running on the processor, and an external host, using the debug interface. When this communication is for debug purposes, it is called the Debug Comms Channel. In an ARMv6 compliant processor, the communications channel includes the Data Transfer Register, some bits of the Data Status and Control Register, and the external debug interface controller, such as the DBGTAP controller in the case of the JTAG interface.

Condition field A four-bit field in an instruction that specifies a condition under which the instruction can execute.

Conditional execution If the condition code flags indicate that the corresponding condition is true when the instruction starts executing, it executes normally. Otherwise, the instruction does nothing.

Context The environment that each process operates in for a multitasking operating system. In ARM processors, this is limited to mean the physical address range that it can access in memory and the associated memory access permissions.

See also Fast context switch.

Control bits The bottom eight bits of a Program Status Register. The control bits change when an exception arises and can be altered by software only when the processor is in a privileged mode.

Core A core is that part of a processor that contains the ALU, the datapath, the general-purpose registers, the Program Counter, and the instruction decode and control circuitry.

Core reset *See* Warm reset.

CoreSight The infrastructure for monitoring, tracing, and debugging a complete system on chip.

Cross Trigger Interface (CTI)
Part of an Embedded Cross Trigger device. The CTI provides the interface between a core/ETM and the CTM within an ECT.

Cross Trigger Matrix (CTM)
The CTM combines the trigger requests generated from CTIs and broadcasts them to all CTIs as channel triggers within an Embedded Cross Trigger device.

CTI *See* Cross Trigger Interface.

CTM *See* Cross Trigger Matrix.

DBGTAP *See* Debug Test Access Port.

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

ECT *See* Embedded Cross Trigger.

Embedded Cross Trigger (ECT)
The ECT is a modular component to support the interaction and synchronization of multiple triggering events with an SoC.

EmbeddedICE-RT The JTAG-based hardware provided by debuggable ARM processors to aid debugging in real-time.

Embedded Trace Macrocell (ETM)
A hardware macrocell that, when connected to a processor core, outputs instruction and data trace information on a trace port.

The Program Flow Trace macrocell provides processor driven instruction-only trace through a trace port compliant to the ATB protocol.

ETM See *Embedded Trace Macrocell*.

Exception A fault or error event that is considered serious enough to require that program execution is interrupted. Examples include attempting to perform an invalid memory access, external interrupts, and undefined instructions. When an exception occurs, normal program flow is interrupted and execution is resumed at the corresponding exception vector. This contains the first instruction of the interrupt handler to deal with the exception.

Exception service routine
See Interrupt handler.

Exception vector *See* Interrupt vector.

Flat address mapping A system of organizing memory in which each Physical Address contained within the memory space is the same as its corresponding Virtual Address.

Halt mode One of two mutually exclusive debug modes. In halt mode all processor execution halts when a breakpoint or watchpoint is encountered. All processor state, coprocessor state, memory and input/output locations can be examined and altered by the JTAG interface.

See also Monitor debug-mode.

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

IGN *See* Ignore.

Ignore (IGN) Must ignore memory writes.

Illegal instruction An instruction that is architecturally Undefined.

Implementation-defined
The behavior is not architecturally defined, but is defined and documented by individual implementations.

Implementation-specific
The behavior is not architecturally defined, and does not have to be documented by individual implementations. Used when there are a number of implementation options available and the option chosen does not affect software compatibility.

Imprecise tracing A filtering configuration where instruction or data tracing can start or finish earlier or later than expected. Most cases cause tracing to start or finish later than expected.

For example, if **TraceEnable** is configured to use a counter so that tracing begins after the fourth write to a location in memory, the instruction that caused the fourth write is not traced, although subsequent instructions are. This is because the use of a counter in the **TraceEnable** configuration always results in imprecise tracing.

Instrumentation trace A component for debugging real-time systems through a simple memory-mapped trace interface, providing `printf` style debugging.

Interrupt handler A program that control of the processor is passed to when an interrupt occurs.

Interrupt vector One of a number of fixed addresses in low memory, or in high memory if high vectors are configured, that contains the first instruction of the corresponding interrupt handler.

Jazelle architecture The ARM Jazelle architecture extends the Thumb and ARM operating states by adding a Java state to the processor. Instruction set support for entering and exiting Java applications, real-time interrupt handling, and debug support for mixed Java/ARM applications is present. When in Java state, the processor fetches and decodes Java bytecodes and maintains the Java operand stack.

JTAG *See* Joint Test Action Group.

JTAG Access Port (JTAG-AP)
An optional component of the DAP that provides JTAG access to on-chip components, operating as a JTAG master port to drive JTAG chains throughout a SoC.

JTAG-AP *See* JTAG Access Port.

JTAG Debug Port (JTAG-DP)
An optional external interface for the DAP that provides a standard JTAG interface for debug access.

JTAG-DP *See* JTAG Debug Port.

Load/store architecture
A processor architecture where data-processing operations only operate on register contents, not directly on memory contents.

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

Memory bank One of two or more parallel divisions of interleaved memory, usually one word wide, that enable reads and writes of multiple words at a time, rather than single words. All memory banks are addressed simultaneously and a bank enable or chip select signal determines which of the banks

is accessed for each transfer. Accesses to sequential word addresses cause accesses to sequential banks. This enables the delays associated with accessing a bank to occur during the access to its adjacent bank, speeding up memory transfers.

Memory coherency A memory is coherent if the value read by a data read or instruction fetch is the value that was most recently written to that location. Memory coherency is made difficult when there are multiple possible physical locations that are involved, such as a system that has main memory, a write buffer and a cache.

Microprocessor *See* Processor.

Miss *See* Cache miss.

Monitor debug-mode One of two mutually exclusive debug modes. In Monitor debug-mode the processor enables a software abort handler provided by the debug monitor or operating system debug task. When a breakpoint or watchpoint is encountered, this enables vital system interrupts to continue to be serviced while normal program execution is suspended.

See also Halt mode.

Multi-ICE A JTAG-based tool for debugging embedded systems.

Multi-layer An interconnect scheme similar to a cross-bar switch. Each master on the interconnect has a direct link to each slave, The link is not shared with other masters. This enables each master to process transfers in parallel with other masters. Contention only occurs in a multi-layer interconnect at a payload destination, typically the slave.

Multi-master AHB Typically a shared, not multi-layer, AHB interconnect scheme. More than one master connects to a single AMBA AHB link. In this case, the bus is implemented with a set of full AMBA AHB master interfaces. Masters that use the AMBA AHB-Lite protocol must connect through a wrapper to supply full AMBA AHB master signals to support multi-master operation.

Power-on reset *See* Cold reset.

Prefetching In pipelined processors, the process of fetching instructions from memory to fill up the pipeline before the preceding instructions have finished executing. Prefetching an instruction does not mean that the instruction has to be executed.

Processor A processor is the circuitry in a computer system required to process data using the computer instructions. It is an abbreviation of microprocessor. A clock source, power supplies, and main memory are also required to create a minimum complete working computer system.

RealView ICE A system for debugging embedded processor cores using a JTAG interface.

Region A partition of instruction or data memory space.

Remapping Changing the address of physical memory or devices after the application has started executing. This is typically done to permit RAM to replace ROM when the initialization has been completed.

Replicator A replicator enables two trace sinks to be wired together and to operate independently on the same incoming trace stream. The input trace stream is output onto two (independent) ATB ports.

Reserved A field in a control register or instruction format is reserved if the field is to be defined by the implementation, or produces Unpredictable results if the contents of the field are not zero. These fields are reserved for use in future extensions of the architecture or are implementation-specific. All reserved bits not used by the implementation must be written as 0 and read as 0.

Saved Program Status Register (SPSR)
The register that holds the CPSR of the task immediately before the exception occurred that caused the switch to the current mode.

SBO *See* Should Be One.

SBZ *See* Should Be Zero.

SBZP *See* Should Be Zero or Preserved.

Scalar operation A VFP coprocessor operation involving a single source register and a single destination register.

See also Vector operation.

SDF *See* Standard Delay Format.

Set *See* Cache set.

Should Be One (SBO) Should be written as 1 (or all 1s for bit fields) by software. Writing a 0 produces Unpredictable results.

Should Be Zero (SBZ) Should be written as 0 (or all 0s for bit fields) by software. Writing a 1 produces Unpredictable results.

Should Be Zero or Preserved (SBZP)
Should be written as 0 (or all 0s for bit fields) by software, or preserved by writing the same value back that has been previously read from the same field on the same processor.

Sign-Off Model (SOM) An opaque, compiled simulation model generated from a technology specific netlist of an ARM processor, derived after gate level synthesis and timing annotation, that you can use in back-annotated gate-level simulations to prove the function and timing behavior of the device. It enables accurate timing simulation of SoCs and simulation using production test vectors from Automatic Test Pattern Generation (ATPG) tool such as Synopsys TetraMAX. It only supports back-annotation using SDF files. The SOM includes timing information but provides slower simulation than a DSM.

Serial-Wire Debug Port
An optional external interface for the DAP that provides a bidirectional debug interface.

Serial-Wire JTAG (SWJ)
A model whereby a run-control emulator (based on RVI-ME) is placed in the chip and communicated with using a single pin scheme (compared to the four to six for JTAG). This not only reduces pins, but SWJ provides power from the run-control emulator (through the pin). It also provides additional access and a unique ID. The use of this DBT model enables this mode to run very fast for download.

SOM *See* Sign-Off Model.

SPICE Simulation Program with Integrated Circuit Emphasis. An accurate transistor-level electronic circuit simulation tool that can predict how an equivalent real circuit behaves for given circuit conditions.

SPSR *See* Saved Program Status Register

Standard Delay Format (SDF)
The format of a file that contains timing information to the level of individual bits of buses and is used in SDF back-annotation. An SDF file can be generated in a number of ways, but most commonly from a delay calculator.

SW-DP *See* Serial-Wire Debug Port.

SWJ *See* Serial-Wire JTAG.

Synchronization primitive
The memory synchronization primitive instructions are those instructions that are used to ensure memory synchronization. That is, the LDREX, STREX, SWP, and SWPB instructions.

TCD *See* Trace Capture Device.

Thumb instruction A halfword that specifies an operation for an ARM processor in Thumb state to perform. Thumb instructions must be halfword-aligned.

Thumb state A processor that is executing Thumb (16-bit) halfword aligned instructions is operating in Thumb state.

TLB *See* Translation Lookaside Buffer.

TPA See *Trace Port Analyzer*.

TPIU *See* Trace Port Interface Unit.

Trace Capture Device (TCD)
A generic term to describe Trace Port Analyzers, logic analyzers, and on-chip trace buffers.

Trace driver A Remote Debug Interface target that controls a piece of trace hardware. That is, the trigger macrocell, trace macrocell, and trace capture tool.

Trace funnel A device that combines multiple trace sources onto a single bus.

Trace hardware A term for a device that contains an Embedded Trace Macrocell.

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

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

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

Translation Lookaside Buffer (TLB)
A cache of recently used page table entries that avoid the overhead of page table walking on every memory access. Part of the Memory Management Unit.

Translation table A table, held in memory, that contains data that defines the properties of memory areas of various fixed sizes.

Translation table walk The process of doing a full translation table lookup. It is performed automatically by hardware.

Unaligned A data item stored at an address that is not divisible by the number of bytes that defines the data size is said to be unaligned. For example, a word stored at an address that is not divisible by four.

Undefined Indicates an instruction that generates an Undefined instruction trap. See the *ARM Architecture Reference Manual* for more details on ARM exceptions.

UNP *See* Unpredictable.

Unpredictable Means that the behavior of the ETM cannot be relied on. Such conditions have not been validated. When applied to the programming of an event resource, only the output of that event resource is Unpredictable.Unpredictable behavior can affect the behavior of the entire system, because the ETM is capable of causing the processor to enter debug state, and external outputs can be used for other purposes.

Warm reset Also known as a core reset. Initializes the majority of the processor excluding the debug controller and debug logic. This type of reset is useful if you are using the debugging features of a processor.

Watchpoint A watchpoint is a mechanism provided by debuggers to halt program execution when the data contained by a particular memory address is changed. Watchpoints are inserted by the programmer to enable inspection of register contents, memory locations, and variable values when memory is written to test that the program is operating correctly. Watchpoints are removed after the program is successfully tested. *See also* Breakpoint.

WB *See* Write-back.

Word A 32-bit data item.

Word-invariant In a word-invariant system, the address of each byte of memory changes when switching between little-endian and big-endian operation, in such a way that the byte with address A in one endianness has address A EOR 3 in the other endianness. As a result, each aligned word of memory always consists of the same four bytes of memory in the same order, regardless of endianness. The change of endianness occurs because of the change to the byte addresses, not because the bytes are rearranged.The ARM architecture supports word-invariant systems in ARMv3 and later versions. When word-invariant support is selected, the behavior of load or store instructions that are given unaligned addresses is instruction-specific, and is in general not the expected behavior for an unaligned access. It is recommended that word-invariant systems use the endianness that produces the desired byte addresses at all times, apart possibly from very early in their reset handlers before they have set up the endianness, and that this early part of the reset handler must use only aligned word memory accesses.

See also Byte-invariant.

Write Writes are defined as operations that have the semantics of a store. That is, the ARM instructions `SRS`, `STM`, `STRD`, `STC`, `STRT`, `STRH`, `STRB`, `STRBT`, `STREX`, `SWP`, and `SWPB`, and the Thumb instructions `STM`, STR, `STRH`, `STRB`, and `PUSH`.

Java instructions that are accelerated by hardware can cause a number of writes to occur, according to the state of the Java stack and the implementation of the Java hardware acceleration.

Write-back (WB) In a write-back cache, data is only written to main memory when it is forced out of the cache on line replacement following a cache miss. Otherwise, writes by the processor only update the cache. This is also known as copyback.

Write buffer A block of high-speed memory, arranged as a FIFO buffer, between the data cache and main memory, whose purpose is to optimize stores to main memory.

Write completion The memory system indicates to the processor that a write has been completed at a point in the transaction where the memory system is able to guarantee that the effect of the write is visible to all processors in the system. This is not the case if the write is associated with a memory synchronization primitive, or is to a Device or Strongly Ordered region. In these cases the memory system might only indicate completion of the write when the access has affected the state of the target, unless it is impossible to distinguish between having the effect of the write visible and having the state of target updated.

This stricter requirement for some types of memory ensures that any side-effects of the memory access can be guaranteed by the processor to have taken place. You can use this to prevent the starting of a subsequent operation in the program order until the side-effects are visible.

Write-through (WT) In a write-through cache, data is written to main memory at the same time as the cache is updated.

WT *See* Write-through.