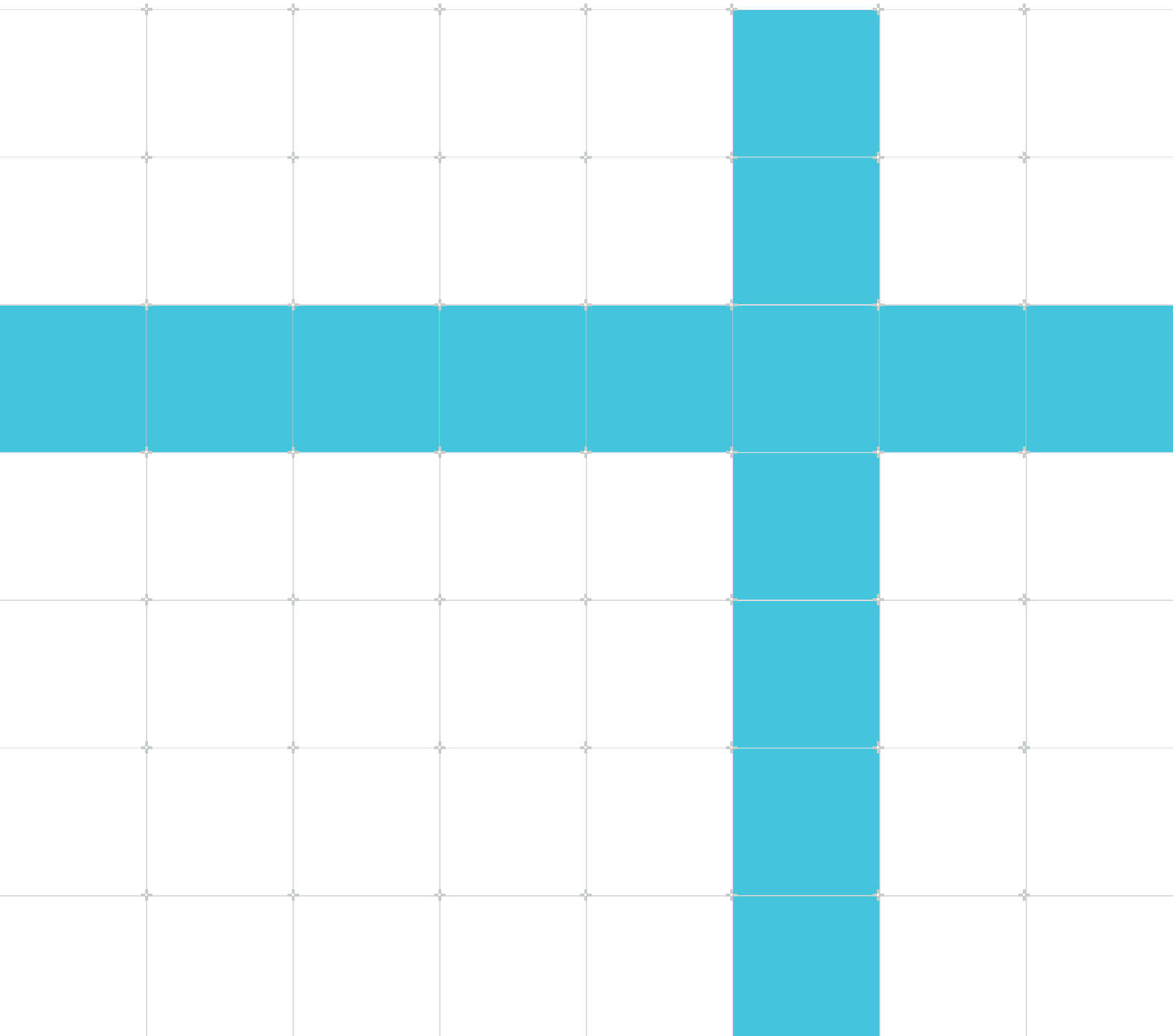
Arm® CoreSight™ ETM-M85

Revision: r0p2

Technical Reference Manual

Issue 05
101926_0002_05_en



Arm® CoreSight™ ETM-M85

Technical Reference Manual

Release Information

Document history

Issue	Date	Confidentiality	Change
0000-02	8 December 2020	Confidential	First beta release for r0p0
0000-03	30 June 2021	Confidential	First limited access release for r0p0
0001-04	15 November 2021	Confidential	First early access release for r0p1
0002-05	26 April 2022	Non-Confidential	First release for r0p2

Proprietary Notice

REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

Feedback

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

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

Inclusive language commitment

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

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

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

Contents

1 Preface

1.1 About this book

Information about the revision status and the intended audience for the book.

1.1.1 Product revision status

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

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

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

1.1.2 Intended audience

This book is written for designers of development tools providing support for ETM functionality and hardware and software engineers integrating the macrocell into an ASIC that includes the Cortex®-M85 processor. Implementation-specific behavior is described in this document. You can find complementary information in the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* and *Arm® Cortex®-M85 Processor Integration and Implementation Manual*.

Using this book

Information about the chapters in this book.

This book is organized into the following chapters:

1 Introduction

2 Functional Description

3 Programmers model

4 ETM-M85 register descriptions

Glossary

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

See the Arm Glossary for more information.

Typographic conventions

The typographic conventions used throughout this document are as follows.

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

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

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

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

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

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

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

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

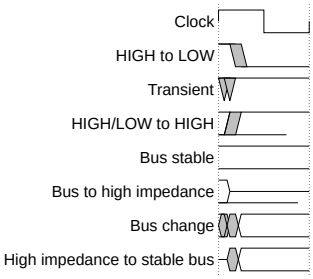
SMALL CAPITALS
: Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

Timing diagrams

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

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

Figure 1-1: Key to timing diagram conventions



Signals

The signal conventions are:

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

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

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

1.1.4 Additional reading

Information published by Arm and by third parties.

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

Arm publications

- *Arm® CoreSight™ DAP-Lite Technical Reference Manual* (DDI 0316)
- *Arm® CoreSight™ Architecture Specification v3.0* (IHI 0029)
- *AMBA® APB Protocol Version 2.0 Specification* (IHI 0033)

- *AMBA® ATB Protocol Specification* (IHI 0032)
- *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* (ARM IHI 0064)
- *Arm®v8-M Architecture Reference Manual* (DDI 0553)
- *Arm® Cortex®-M85 Processor Technical Reference Manual* (101924)
- *Arm® PMC-100 Technical Reference Manual* (101528)

The following confidential book is only available to licensees:

- *Arm® Cortex®-M85 Processor Integration and Implementation Manual* (101925)

Other publications

None.

1.2 Feedback

Arm welcomes feedback on this product and its documentation.

1.2.1 Feedback on this product

Information about how to give feedback on the 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.

1.2.2 Feedback on content

Information about how to give feedback on the content.

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

- The title Arm® CoreSight™ ETM-M85 Technical Reference Manual.
- The number 101926_0002_05_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.


Note

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

2 Introduction

This chapter describes ETM-M85.

2.1 CoreSight™ ETM-M85 features

ETM-M85 can provide non-intrusive program-flow trace for the Cortex®-M85 processor. ETM-M85 generates information that trace software tools use to reconstruct the execution of an entire program or part of a program.

ETM-M85 implements instruction trace only, and can trace:

- All instructions, including condition code pass or fail
- Target addresses of indirect branch operations that have been taken
- Target addresses of direct branch operations that have been taken when TRCCONFIGR.BB is set to 1
- Exceptions
- Entry to debug state when Halting debug mode is enabled
- Cycle counts relating to instruction execution

ETM-M85 contains resource logic that enables you to control instruction trace. This resource logic includes one reduced function counter. For more information on the provision of a reduced function counter, see the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*. Other resources available for ETM-M85 include *Data Watchpoint and Trace*(DWT) processor comparators and external inputs. You can specify the exact set of trigger and filter conditions that are required for a particular application.

For more information about CoreSight™, see:

- *Arm® CoreSight™ Architecture Specification v3.0*
- *Arm® CoreSight™ System-on-Chip SoC-600 Technical Reference Manual*

For more information about the ETM architecture, see the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*.

2.1.1 The CoreSight™ debug environment

The CoreSight™ debug environment contains a software debugger that provides a user interface to ETM-M85. ETM-M85 is designed for instruction trace and it has a single 8-bit AMBA® 4 ATB interface.

Software debugger

A software debugger provides a user interface to ETM-M85. You can use this interface to:

- Configure ETM-M85 facilities such as filtering.
- Configure optional trace features such as cycle counting.
- Configure the other CoreSight™ components such as the *Trace Port Interface Unit* (TPIU).
- Access the processor debug registers and *Performance Monitoring Units* (PMUs).

ETM-M85 outputs its trace to the AMBA® 4 ATB interface.

You can use the CoreSight™ infrastructure to design systems that provide the option to:

- Export the trace information through a trace port. An external *Trace Port Analyzer* (TPA) captures the trace information as shown in Figure 2-1: Example CoreSight system with ETM-M85 on page 15.
- Write the trace information to a trace-capable device that can access local or system memory. You can read out the trace at low speed using a *Joint Test Action Group* (JTAG) or *Serial Wire* (SW) interface.

The software debugger has a copy of the executed image from memory and the captured trace information from the TPA or on-chip trace buffer. It decompresses the image to provide full disassembly with symbols of the code that was executed. ETM-M85 generates trace information that gives the software debugger the capability to link this data back to the original high-level source code. This information provides a visualization of how the code was executed on the Cortex®-M85 processor.

Example CoreSight™ system with ETM-M85

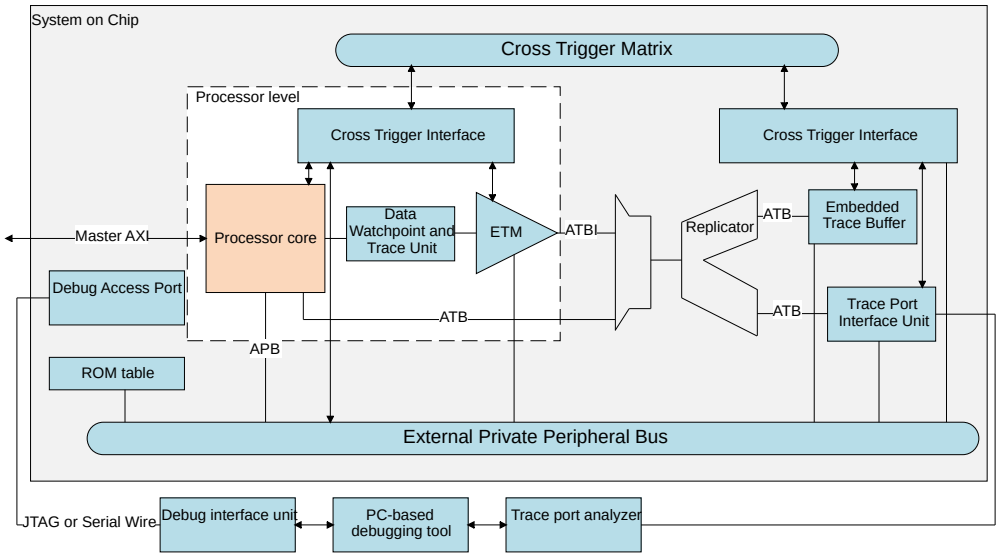
The following figure shows an example of how ETM-M85 fits into a CoreSight™ debug environment to provide instruction trace capabilities in a single processor system.

In this example, the external debug software configures the trace and debug components through the *Debug Access Port* (DAP). The top-level ROM table contains:

- A unique identification code for the SoC.
- The base addresses of the components that are connected to the *External Private Peripheral Bus* (EPPB) on the Cortex®-M85 processor.

The ETM-M85 trace interfaces are replicated to provide on-chip storage using the CoreSight™ ETB and output off-chip using the TPIU. Cross-triggering operates through the *Cross Trigger Interface* (CTI) and *Cross Trigger Matrix* (CTM) components.

Figure 2-1: Example CoreSight™ system with ETM-M85



As an alternative to using an external computer to run a software debugger, the Cortex®-M85 processor (or another processor on the *System-on-Chip* (SoC) can access ETM-M85 and an on-chip trace buffer to provide self-hosted debug and trace functionality.

2.2 Compliance

ETM-M85 is compatible with the CoreSight™ architecture.

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

The ETM-M85 implements the architecture specification and protocols shown in the following table.

Table 2-1: Architecture specifications and protocols

Component	Specification
ETM architecture	ETM-M85 implements the ETM architecture version 4.5. For more information, see *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*.
Interconnect architecture	ETM-M85 complies with the AMBA® 4 APB and AMBA® 4 ATB protocols. For more information, see the *AMBA® APB Protocol Version 2.0 Specification* and *AMBA® ATB Protocol Specification*.

2.3 Features

The ETM-M85 consists of a range of **IMPLEMENTATION DEFINED** features and also implements several optional features from the ETM architecture.

See the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* for information about:

- The trace protocol
- The ETM version 4.5 features
- Controlling tracing using triggering and filtering resources

The following table shows the ETM-M85 features that are **IMPLEMENTATION DEFINED**, in terms of either:

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

Table 2-2: ETM-M85 features with IMPLEMENTATION DEFINED number of instances or size

Feature	ETM-M85 configuration	Description
Address comparators	0 pairs	See bits[3:0] of the 5.16.5 TRCIDR4, ID Register 4 on page 55.
Instruction trace cycle counting minimum allowable threshold value	4	-
Data value comparators	0	See bits[7:4] of the 5.16.5 TRCIDR4, ID Register 4 on page 55.
Context ID comparators	0	See bits[27:24] 5.16.5 TRCIDR4, ID Register 4 on page 55.
Single-Shot comparator controls	1	The single-shot comparators are only sensitive to the processor comparator inputs. See bits [23:20] in 5.16.5 TRCIDR4, ID Register 4 on page 55.
Counters	1	This indicates reduced function counter implementation. See bits[30:28] of the 5.16.6 TRCIDR5, ID Register 5 on page 57.
Trace events supported	2	See bits[11:10] of the 5.16.1 TRCIDR0, ID Register 0 on page 50.
Cycle counter size	12 bits	See bits[28:25] of the 5.16.3 TRCIDR2, ID Register 2 on page 53.
Sequencer	0	See bits[27:25] of the 5.16.6 TRCIDR5, ID Register 5 on page 57.
Processor comparator inputs	4 or 8	See bits[15:12] of the 5.16.5 TRCIDR4, ID Register 4 on page 55.
External inputs	4+number of *Performance Monitor Unit* (PMU) events	See bits[8:0] of the 5.16.6 TRCIDR5, ID Register 5 on page 57.
External outputs	2	-
External input selectors	4	See bits[11:9] of the 5.16.6 TRCIDR5, ID Register 5 on page 57.

Feature	ETM-M85 configuration	Description
Resource selection pairs	2	See bits[19:16] of the 5.16.5 TRCIDR4, ID Register 4 on page 55.
Instruction trace port size	8-bit	-
Instruction FIFO	64 bytes with 8-bit output	Uses *Advanced Trace Bus* (ATB).
Claim tag bits	4	-

The following table shows the optional features of the ETM architecture that ETM-M85 implements.

Table 2-3: ETM-M85 implementation of optional features

Feature	Implemented	Description
Commit mode	No	-
Configurable FIFO	No	-
Trace Start/Stop block	Yes	-
Branch broadcast tracing support	Yes	See bit[5] of the 5.16.1 TRCIDR0, ID Register 0 on page 50.
Trace of conditional instructions	Yes	See bits[13:12] and bit[6] of the 5.16.1 TRCIDR0, ID Register 0 on page 50.
Data trace	No	-
System error trace	Yes	-
Load and store instruction data trace	No	-
Low-power override	Yes	See bit[23] of the 5.16.6 TRCIDR5, ID Register 5 on page 57.
Support for overflow avoidance	Yes	-
Cycle counting in instruction trace	Yes	See bit[7] of the 5.16.1 TRCIDR0, ID Register 0 on page 50.
Data address comparison	No	ETM-M85 does not implement data address comparison.
OS Lock mechanism	No	The Cortex®-M85 processor does not implement OS Lock.
Secure non-invasive debug	Yes	The Cortex®-M85 processor implements optional Security Extensions.
Context ID tracing	No	See bits[9:5] of the 5.16.3 TRCIDR2, ID Register 2 on page 53.
Trace output	Yes	Uses ATB.
Timestamp size	64-bit	See bits[28:24] of the 5.16.1 TRCIDR0, ID Register 0 on page 50.
Memory mapped access to ETM-M85 registers	Yes	See the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* for more information about the access permissions behavior on register accesses for different states of ETM-Cortex®-M85.
External debugger access to ETM-M85 registers	Yes	
System instruction access to ETM-M85 registers	No	

Feature	Implemented	Description
Virtual Machine ID (VMID) comparator support	No	See bits[31:28] of the 5.16.5 TRCIDR4, ID Register 4 on page 55.
Q-element filtering support	No	-
Q-element support	No	-
Reduced function counter	Yes	See bit [31] in 5.16.6 TRCIDR5, ID Register 5 on page 57.
Return stack support	Yes	-
Stall control support	Yes	-
Software lock	No	-
Synchronization period support	Yes	Read-only support is provided
ATB trigger support	Yes	See bit[22] of the 5.16.6 TRCIDR5, ID Register 5 on page 57.

2.4 Interfaces and configurable options

ETM-M85 has four main interfaces.

ETM-M85 interfaces

The following table shows the main interfaces in ETM-M85.

Table 2-4: ETM-M85 interfaces

Interface	Description
Processor interface	This interface connects the M85 processor to ETM-M85. It has the following functions: • Tracking execution information from the processor. • Decoding the control signals. • Passing on the decoded information to the internal interfaces.
AMBA® 4 ATB interface	This interface is the instruction ATB interface. This interface reads single bytes of packet information from the instruction FIFO and the information is sent over the interface.
AMBA® 4 APB interface	This interface is an interface to the APB that provides access to the programmable registers.
Production test interface	This interface contains the scan enable signal that is used in ETM-M85 production testing.

Configurable options

ETM-M85 has no configurable options.

2.5 Design process

ETM-M85 is delivered as synthesizable RTL.

Before it can be used in a product, ETM-M85 must go through the design process described in the following table.

Table 2-5: ETM-M85 design process

Stage	Description
Implementation	For *Microcontroller Units* (MCUs), often a single design team integrates the processor and ETM-M85 before synthesizing the complete design. Alternatively, the team can synthesize the processor on its own or partially integrated, to produce a hard macrocell. Then, the same team or a separate team, integrates the hard macrocell into the design.
Integration	The integrator connects the implemented design into a *System on Chip* (SoC). This includes connecting it to a memory system and peripherals.
Programming	This is the last stage in the process. The system programmer develops the software required to configure and initialize ETM-M85, and tests the required application software.

The operation of the final device depends on:

Build configuration

The implementer chooses the options that affect how the RTL source files are pre-processed. These options usually include or exclude logic that affects one or more of the following:

- Area.
- Maximum frequency.
- Features of the resulting design.

Configuration inputs

The integrator configures some of the ETM-M85 features by tying inputs to specific values. These configurations affect the start-up behavior before any software configuration is made. They can also limit the options available to the software.

Software configuration

The programmer configures ETM-M85 by programming particular values into registers. These register values affect the ETM-M85 behavior.


Note

- This manual refers to **IMPLEMENTATION DEFINED** features that are applicable to build configuration options. Any reference to:
  - An included feature means that the appropriate build and pin configuration options are selected.
  - An enabled feature means one that has also been configured by software.
- Each stage of the process:
  - Can be performed by a different party.
  - Can include implementation and integration choices that affect the ETM-M85 behavior and features.

2.6 Documentation

ETM-M85 documentation is as follows:

Technical Reference Manual

The *Technical Reference Manual* (TRM) describes the functionality and the effects of functional options on the ETM-M85 behavior. 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 ETM-M85 is implemented and integrated.

Integration and Implementation Manual

For both the processor and ETM-M85, the *Arm® Cortex®-M85 Processor Integration and Implementation Manual* (IIM) describes:

- The available build configuration options and related issues in selecting them.
- How to configure the *Register Transfer Level* (RTL) with the build configuration options.
- How to integrate the processor into a *System on Chip* (SoC). This includes a description of the pins that the integrator must tie off to configure the macrocell for the required integration.
- The processes to sign off the integration and implementation of the 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 IIM.

The IIM is a confidential document that is only available to licensees.

2.7 Product revisions

The following product revisions have been released.

r0p0	First beta release for r0p0
r0p1	First early access release for r0p1
r0p2	First release for r0p2

3 Functional Description

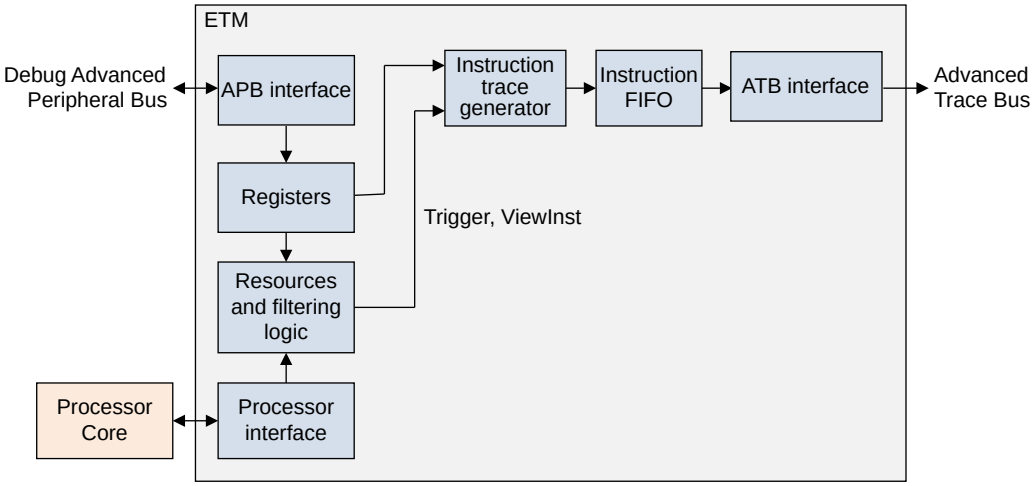
This chapter describes the ETM-M85 functional features and operation.

3.1 ETM-M85 functional blocks

ETM-M85 is a CoreSight™ component and an integral part of the Arm real-time debug solution, Arm Development Studio. ETM-M85 performs real-time instruction tracing based on the ETM version 4.5 architecture.

The following figure shows the main functional blocks of ETM-M85.

Figure 3-1: ETM-M85 block diagram



The following table describes the ETM-M85 functional blocks.

Table 3-1: ETM-M85 functional blocks

Functional block	Description
Processor interface	The processor interface from ETM-M85 connects to the Cortex®-M85 processor. It has the following functions: • Tracking execution information from the processor. • Decoding the control signals. • Passing on the decoded information to the internal interfaces.
Resources and filtering logic	These blocks contain resources which are programmed by trace software to trigger and filter the trace information. They start and stop trace generation, depending on the conditions that have been set.
AMBA® 4 APB interface	This is an interface to the APB that provides access to the programmable registers.

Functional block	Description
Instruction trace generator	This block generates the trace packets that are a compressed form of the instruction execution information provided by the Cortex®-M85 processor. The trace packets are then passed to the instruction FIFO.
Instruction FIFO	This block buffers bursts of trace packets. There is one FIFO provided for the instruction trace stream.
AMBA® 4 ATB interface	This is the instruction ATB interface. This interface reads single bytes of packet information from the instruction FIFO and the information is sent over the interface.
Global timestamping	ETM-M85 supports connection to a global timestamp source. This provides a 64-bit timestamp that a debugger can use for coarse-grained profiling and correlation of the trace source. Arm recommends that the timestamp counter is no slower than 10% of the maximum processor clock frequency.

3.2 External input and output connections

The following table summarizes the external input connections. These inputs can be selected using the external input selectors controlled by TRCEXTINSELR.

For more information, see 5.14 TRCEXTINSELR, External Input Select Register on page 48.

Table 3-2: External input connections

External input bits	Description
[3:0]	*Cross Trigger Interface* (CTI)
[3+N:4] where N is the number of PMU events	*Performance Monitoring Unit* (PMU) events

The following table shows the ETM-M85 external input and output bit connections to the *Cross Trigger Interface* (CTI).

Table 3-3: CTI connections

ETM-M85 external input and output bits	CTI input
ETM external input bits [3:0]	CTI output bits [7:4]
ETM external output bits [1:0]	CTI input bits [5:4]

The following table shows the ETM-M85 external input and output bit connections to the *Performance Monitoring Unit* (PMU).

Table 3-4: PMU events

Event number	Event mnemonic	External input bit	Event name
0x0000	SW_INCR	4	Instruction architecturally executed, condition code check pass, software increment
0x0001	L1I_CACHE_REFILL	5	L1 instruction cache linefill
0x0003	L1D_CACHE_REFILL	6	L1 data cache linefill
0x0004	L1D_CACHE	7	L1 data cache access

Event number	Event mnemonic	External input bit	Event name
0x0006	LD_RETIRED	8	Instruction architecturally executed, condition code check pass, load
0x0007	ST_RETIRED	9	Instruction architecturally executed, condition code check pass, store
0x0008	INST_RETIRED	10	Instruction architecturally executed
0x0009	EXC_TAKEN	11	Exception taken
0x000A	EXC_RETURN	12	Instruction architecturally executed, condition code check pass, exception return
0x000C	PC_WRITE_RETIRED	13	Instruction architecturally executed, condition code check pass, software change of the PC
0x000D	BR_IMMED_RETIRED	14	Instruction architecturally executed, immediate branch
0x000E	BR_RETURN_RETIRED	15	Instruction architecturally executed, condition code check pass, procedure return
0x000F	UNALIGNED_LDST_RETIRED	16	Instruction architecturally executed, condition code check pass, unaligned load or store
0x0010	BR_MIS_PRED	17	Mispredicted or not predicted branch speculatively executed
0x0011	CPU_CYCLES	18	Cycle
0x0012	BR_PRED	19	Predictable branch speculatively executed
0x0013	MEM_ACCESS	20	Data memory access
0x0014	L1I_CACHE	21	L1 instruction cache access
0x0015	L1D_CACHE_WB	22	L1 data cache write-back
0x0019	BUS_ACCESS	23	Any beat access to the M-AXI read interface, M-AXI write interface and any access to P-AHB interface or *External Private Peripheral Bus* (EPPB) interface
0x001A	MEMORY_ERROR	24	ECC error for *Tightly Coupled Memories* (TCMs) and caches
0x001D	BUS_CYCLES	26	AXI Bus cycle when ACKEN is asserted
0x001E	CHAIN	27	For an odd-numbered counter, increments when an overflow occurs on the preceding even-numbered counter on the same PE
0x0021	BR_RETIRED	29	Instruction architecturally executed, branch
0x0022	BR_MIS_PRED_RETIRED	30	Instruction architecturally executed, mispredicted branch
0x0023	STALL_FRONTEND	31	If there are no instructions available from the fetch stage of the processor pipeline, the processor considers the front-end of the processor pipeline as being stalled
0x0024	STALL_BACKEND	32	If there is an instruction available from the fetch stage of the pipeline but it cannot be accepted by the decode stage of the processor pipeline, the processor considers the back-end of the processor pipeline as being stalled
0x0036	LL_CACHE_RD	33	L1 data cache read For the Cortex®-M85 processor, this event is the same as L1D_CACHE_RD

Event number	Event mnemonic	External input bit	Event name
0x0037	LL_CACHE_MISS_RD	34	L1 data cache read miss For the Cortex®-M85 processor, this event is the same as L1D_CACHE_MISS_RD
0x0039	L1D_CACHE_MISS_RD	35*	L1 data cache read miss For the Cortex®-M85 processor, this event is the same as LL_CACHE_MISS_RD
0x003C	STALL	38	No operation sent for execution
0x0040	L1D_CACHE_RD	42	L1 data cache read For the Cortex®-M85 processor, this event is the same as LL_CACHE_RD
0x0100	LE_RETIRED	42	Loop end instruction architecturally executed, entry registered in the LO_BRANCH_INFO cache
0x0108	LE_CANCEL	47	LO_BRANCH_INFO cache containing a valid loop entry cleared while not in the last iteration of the loop
0x0114	SE_CALL_S	49	Call to secure function, resulting in security state change
0x0115	SE_CALL_NS	50	Call to Non-secure function, resulting in security state change
0x0118	DWT_CMPMATCH0	51	*Data Watchpoint and Trace* (DWT) comparator 0 match
0x0119	DWT_CMPMATCH1	52	DWT comparator 1 match
0x011A	DWT_CMPMATCH2	53	DWT comparator 2 match
0x011B	DWT_CMPMATCH3	54	DWT comparator 3 match
0x011C	DWT_CMPMATCH4	145	DWT comparator 4 match
0x011D	DWT_CMPMATCH5	146	DWT comparator 5 match
0x011E	DWT_CMPMATCH6	147	DWT comparator 6 match
0x011F	DWT_CMPMATCH7	148	DWT comparator 7 match
0x0200	MVE_INST_RETIRED	55	*M-profile Vector Extension* (MVE) instruction architecturally executed
0x0204	MVE_FP_RETIRED	57	MVE floating-point instruction architecturally executed
0x0208	MVE_FP_HP_RETIRED	59	MVE half-precision floating-point instruction architecturally executed
0x020C	MVE_FP_SP_RETIRED	61	MVE single-precision floating-point instruction architecturally executed
0x0214	MVE_FP_MAC_RETIRED	63	MVE floating-point multiply or multiply accumulate instruction architecturally executed
0x0224	MVE_INT_RETIRED	65	MVE integer instruction architecturally executed
0x0228	MVE_INT_MAC_RETIRED	67	MVE integer multiply or multiply-accumulate instruction architecturally executed
0x0238	MVE_LDST_RETIRED	69	MVE load or store instruction architecturally executed
0x023C	MVE_LD_RETIRED	71	MVE load instruction architecturally executed
0x0240	MVE_ST_RETIRED	73	MVE store instruction architecturally executed

Event number	Event mnemonic	External input bit	Event name
0x0244	MVE_LDST_CONTIG_RETIRED	75	MVE contiguous load or store instruction architecturally executed
0x0248	MVE_LD_CONTIG_RETIRED	77	MVE contiguous load instruction architecturally executed
0x024C	MVE_ST_CONTIG_RETIRED	79	MVE contiguous store instruction architecturally executed
0x0250	MVE_LDST_NONCONTIG_RETIRED	81	MVE non-contiguous load or store instruction architecturally executed
0x0254	MVE_LD_NONCONTIG_RETIRED	83	MVE non-contiguous load instruction architecturally executed
0x0258	MVE_ST_NONCONTIG_RETIRED	85	MVE non-contiguous store instruction architecturally executed
0x025C	MVE_LDST_MULTI_RETIRED	87	MVE memory instruction targeting multiple registers architecturally executed
0x0260	MVE_LD_MULTI_RETIRED	89	MVE memory load instruction targeting multiple registers architecturally executed
0x0264	MVE_ST_MULTI_RETIRED	91	MVE memory store instruction targeting multiple registers architecturally executed
0x028C	MVE_LDST_UNALIGNED_RETIRED	93	MVE unaligned memory load or store instruction architecturally executed
0x0290	MVE_LD_UNALIGNED_RETIRED	95	MVE unaligned load instruction architecturally executed
0x0294	MVE_ST_UNALIGNED_RETIRED	97	MVE unaligned store instruction architecturally executed
0x0298	MVE_LDST_UNALIGNED_NONCONTIG_RETIRED	99	MVE unaligned non-contiguous load or store instruction architecturally executed
0x02A0	MVE_VREDUCE_RETIRED	101	MVE vector reduction instruction architecturally executed
0x02A4	MVE_VREDUCE_FP_RETIRED	103	MVE floating-point vector reduction instruction architecturally executed
0x02A8	MVE_VREDUCE_INT_RETIRED	105	MVE integer vector reduction instruction architecturally executed
0x02B8	MVE_PRED	106	Cycles where one or more predicated beats architecturally executed
0x02CC	MVE_STALL	107	Stall cycles caused by an MVE instruction
0x02CD	MVE_STALL_RESOURCE	108	Stall cycles caused by an MVE instruction because of resource conflicts
0x02CE	MVE_STALL_RESOURCE_MEM	109	Stall cycles caused by an MVE instruction because of memory resource conflicts
0x02CF	MVE_STALL_RESOURCE_FP	110	Stall cycles caused by an MVE instruction because of floating-point resource conflicts
0x02D0	MVE_STALL_RESOURCE_INT	111	Stall cycles caused by an MVE instruction because of integer resource conflicts
0x02D3	MVE_STALL_BREAK	112	Stall cycles caused by an MVE chain break
0x02D4	MVE_STALL_DEPENDENCY	113	Stall cycles caused by MVE register dependency
0x4007	ITCM_ACCESS	114	*Instruction Tightly Coupled Memory* (ITCM) access
0x4008	DTCM_ACCESS	115	*Data Tightly Coupled Memory* (DTCM) access
0x4010	TRCEXTOUT0	116	*Embedded Trace Macrocell* (ETM) external output 0
0x4011	TRCEXTOUT1	117	ETM external output 1
0x4012	TRCEXTOUT2	118	ETM external output 2
0x4013	TRCEXTOUT3	119	ETM external output 3

Event number	Event mnemonic	External input bit	Event name
0x4018	CTI_TRIGOUT4	120	*Cross Trigger Interface* (CTI) output trigger 4
0x4019	CTI_TRIGOUT5	121	CTI output trigger 5
0x401A	CTI_TRIGOUT6	122	CTI output trigger 6
0x401B	CTI_TRIGOUT7	123	CTI output trigger 7
0xC000	ECC_ERR	124	One or more *Error Correcting Code* (ECC) errors detected
0xC001	ECC_ERR_MBIT	125	One or more multi-bit ECC errors detected
0xC010	ECC_ERR_DCACHE	125	One or more ECC errors in the data cache
0xC011	ECC_ERR_ICACHE	127	One or more ECC errors in the instruction cache
0xC012	ECC_ERR_MBIT_DCACHE	128	One or more multi-bit ECC errors in the data cache
0xC013	ECC_ERR_MBIT_ICACHE	129	One or more multi-bit ECC errors in the instruction cache
0xC020	ECC_ERR_DTCM	130	One or more ECC errors in the *Data Tightly Coupled Memory* (DTCM)
0xC021	ECC_ERR_ITCM	131	One or more ECC errors in the *Instruction Tightly Coupled Memory* (ITCM)
0xC022	ECC_ERR_MBIT_DTCM	132	One or more multi-bit ECC errors in the DTCM
0xC023	ECC_ERR_MBIT_ITCM	133	One or more multi-bit ECC errors in the ITCM
0xC100	PF_LINEFILL	134	The prefetcher starts a linefill
0xC101	PF_CANCEL	135	The prefetcher stops prefetching
0xC102	PF_DROP_LINEFILL	136	A linefill triggered by the prefetcher has been dropped because of a lack of buffering
0xC200	NWAMODE_ENTER	137	No-write allocate mode entry
0xC201	NWAMODE	138	Write-Allocate store is not allocated into the data cache due to no-write-allocate mode
0xC300	SAHB_ACCESS	139	Read or write access on the S-AHB interface to the TCM
0xC301	PAHB_ACCESS	140	Read or write access to the P-AHB write interface
0xC302	AXI_WRITE_ACCESS	141	Any beat access to M-AXI write interface.
0xC303	AXI_READ_ACCESS	142	Any beat access to M-AXI read interface
0xC400	DOSTIMEOUT_DOUBLE	143	Denial of Service timeout has fired twice and caused buffers to drain to allow forward progress
0xC401	DOSTIMEOUT_TRIPLE	144	Denial of Service timeout has fired three times and blocked the LSU to force forward progress
0xC500	FUSED_INST_RETIRED	149	Fused instructions architecturally executed
0xC501	BR_INDIRECT	150	Indirect branch instruction architecturally executed
0xC502	BTAC_HIT	151	BTAC branch predictor hit
0xC503	BTAC_HIT_RETURNS	152	Return branch hits BTAC
0xC504	BTAC_HIT_CALLS	153	Call branch hits BTAC
0xC505	BTAC_HIT_INDIRECT	154	Indirect branch hits BTACT
0xC506	BTAC_NEW_ALLOC	155	New allocation to BTAC
0xC507	BR_IND_MIS_PRED	156	Indirect branch mis-predicted
0xC508	BR_RETURN_MIS_PRED	157	Return branch mis-predicted
0xC509	BR_BTAC_OFFSET_OVERFLOW	158	Branch does not allocate in BTAC due to offset overflow

Event number	Event mnemonic	External input bit	Event name
0xC50A	STB_FULL_STALL_AXI	159	*STore Buffer* (STB) full with AXI requests causing CPU to stall
0xC50B	STB_FULL_STALL_TCM	160	STB full with TCM requests causing CPU to stall
0xC50C	CPU_STALLED_AHBS	161	CPU is stalled because TCM access through AHBS
0xC50D	AHBS_STALLED_CPU	162	AHBS is stalled due to TCM access by CPU
0xC50E	BR_INTERSTATING_MIS_PRED	163	Inter-stating branch is mis-predicted.
0xC50F	DWT_STALL	164	*Data Watchpoint and Trace* (DWT) stall
0xC510	DWT_FLUSH	165	DWT flush
0xC511	ETM_STALL	166	*Embedded Trace Macrocell* (ETM) stall
0xC512	ETM_FLUSH	167	ETM flush
0xC513	ADDRESS_BANK_CONFLICT	168	Bank conflict prevents memory instruction dual issue
0xC514	BLOCKED_DUAL_ISSUE	169	Dual instruction issuing is prevented
0xC515	FP_CONTEXT_TRIGGER	170	Floating Point Context is created
0xC516	TAIL_CHAIN	171	New exception is handled without first unstacking
0xC517	LATE_ARRIVAL	172	Late-arriving exception taken during exception entry
0xC518	INT_STALL_FAULT	173	Delayed exception entry due to ongoing fault processing
0xC519	INT_STALL_DEV	174	Delayed exception entry due to outstanding device access
0xC51A	PAC_STALL	175	Stall caused by authentication code computation
0xC51B	PAC_RETIRED	176	PAC instruction architecturally executed
0xC51C	AUT_RETIRED	177	AUT instruction architecturally executed
0xC51D	BTI_RETIRED	178	BTI instruction architecturally executed
0xC51E	PF_NL_MODE	179	Prefetch in next line mode
0xC51F	PF_STREAM_MODE	180	Prefetch in stream mode
0xC520	PF_BUFF_CACHE_HIT	181	Prefetch request that hit in the cache
0xC521	PF_REQ_LFB_HIT	182	Prefetch request that hit in line fill buffers
0xC522	PF_BUFF_FULL	183	Number of times prefetch buffer is full
0xC523	PF_REQ_DCACHE_HIT	184	Generated prefetch request address that hit in D-Cache

3.3 Operation

The ETM-M85 has **IMPLEMENTATION DEFINED** operations.

For information on the operation, see the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*.

3.3.1 ETM-M85 registers

There are two groups of ETM-M85 registers:

- Registers that are completely defined by the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*.
- Registers that are partly **IMPLEMENTATION DEFINED**.

3.3.2 Precise ViewInst events

The only condition that ensures **ViewInst** is precise is that the enabling event condition is TRUE.

For more information, see the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*.

3.3.3 Parallel instruction execution

The Cortex®-M85 processor supports parallel instruction execution. This means that ETM-M85 is capable of tracing up to three instructions per cycle.

If **ViewInst** is active in a cycle when multiple instructions are executed, then all the executed instructions are traced.

3.3.4 Trace features

The ETM-M85 implements the following optional ETMv4.5 trace features:

- Cycle-counting
- Timestamping
- Branch broadcasting
- Conditional instruction tracing

See the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* for descriptions of these features.

3.3.5 Packet formats

The packet formats that the ETM-M85 instruction trace interface supports.

The ETM-M85 instruction trace interface does not support the following trace packet types:

- Speculation resolution packets are not supported.
- Q instruction trace packets are not supported.

ETM-M85 supports conditional tracing, but does not trace APSR condition flag values. See the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* for the trace packet format descriptions.

3.3.6 Resource selection

The ETM-M85 uses event selectors to control resources.

The ETM-M85 has the following resources:

- Reduced function counter
- *Data Watchpoint and Trace* (DWT) processor comparators
- Single shot comparator
- External inputs

The ETM-M85 generates the following events:

- Trace events, triggers, and markers in the trace stream
- Timestamp event
- ViewInst event

An event selector is configured to be sensitive to a resource selector pair, and one resource selector pair can control more than one event selector. The event selectors for Cortex®-M85 are located in the following registers:

- Event Control 0 Register, TRCEVENTCTL0R
- Global Timestamp Control Register, TRCTSCTLR
- ViewInst Main Control Register, TRCVICTLR

The ETM-M85 provides:

- A fixed resource selector pair, registers TRCRSCTLR0 and TRCRSCTLR1 with static values of 0 = FALSE and 1 = TRUE, respectively
- A configurable resource selector pair, registers TRCRSCTLR2 and TRCRSCTLR3

A resource selector pair enables up to two resource groups to be selected, and enables one or more resources to be selected in each group. If more than one resource is selected, the outputs of the selected resources are OR-gated. See the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* for more information.

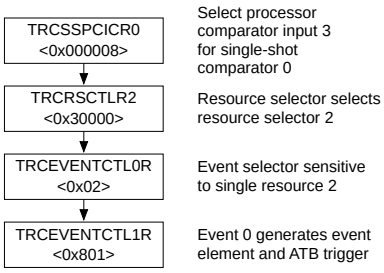
The following table shows the resources that can be selected for the instruction trace.

Table 3-5: Instruction trace resource selection

Group	Select	Resource
0b0000	Bits 0-3	External input selectors 0-3 When select bit N is set, then the resource selector is sensitive to external input selector N.
0b0001	Bits 0-7	When select bit N is set, the resource selector is sensitive to processor comparator input N.
0b0010	0	Counter at zero 0
0b0011	0	Single-Shot Comparator control 0
0b0100-0b1111	0-15	Reserved

The following figure shows the steps necessary to use a single-shot comparator to generate a trigger event and an *Advanced Trace Bus* (ATB) trigger. This example uses the user-configurable resource selector 2.

Figure 3-2: Trigger event resource selection



The DWT unit controls the processor comparator inputs.

3.3.7 Trace flush behavior

Events that ETM-M85 observes can be confirmed to have reached the trace bus output by using the ATB flush protocol.

The ETM-M85 internally flushes instruction trace whenever the flush request is seen. When the processor enters a low-power state, this also causes instruction trace to be output from the ETM-M85.

3.3.8 Low-power state behavior

When the processor enters a low-power state, there is a delay before the resources in the ETM-M85 become inactive.

This delay permits the last instruction executed to trigger a comparator or update the counter, and the resulting event packet to be inserted in the specified trace stream. This event packet is presented on the trace bus before the ETM-M85 enters a low-power state.

If an event packet is generated for a different reason, it is not guaranteed to be output before the ETM-M85 enters a low-power state, but is traced when the processor leaves the low-power state. Reset the ETM-M85 logic before this can occur.

This low-power behavior can be disabled using TRCEVENTCTL1R.LPOVERRIDE bit. For more information on TRCEVENTCTL1R, see 5.6 TRCEVENTCTL1R, Event Control 1 Register on page 40. In this case, the ETM-M85 resources remain active.

3.3.9 Cycle counter

The ETM-M85 uses a 12-bit cycle counter.

The cycle counter does not count when non-invasive debug is disabled or when ETM-M85 is in a low-power state.

3.3.10 Event tracing and triggers

Instruction event packets can be inserted in the instruction trace stream on every cycle. If events are traced continuously on every cycle the instruction FIFO is unable to drain because the rate at which the events enter the FIFO is the same as the rate at which events leave the FIFO. Therefore, the instruction FIFO overflows.

When used with the optimized Cortex®-M85 TPIU, *Advanced Trace Bus* (ATB) triggers must not be enabled, TRCEVENTCTL1R.ATB must be set to 0. For more information on TRCEVENTCTL1R, see 5.6 TRCEVENTCTL1R, Event Control 1 Register on page 40.

4 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-M85 registers to control the macrocell.

4.1 Modes of operation and execution

This section describes how to control and program the ETM and its registers.

4.1.1 Controlling ETM-M85 programming

When programming the ETM-M85 registers, you must enable all the changes at the same time.

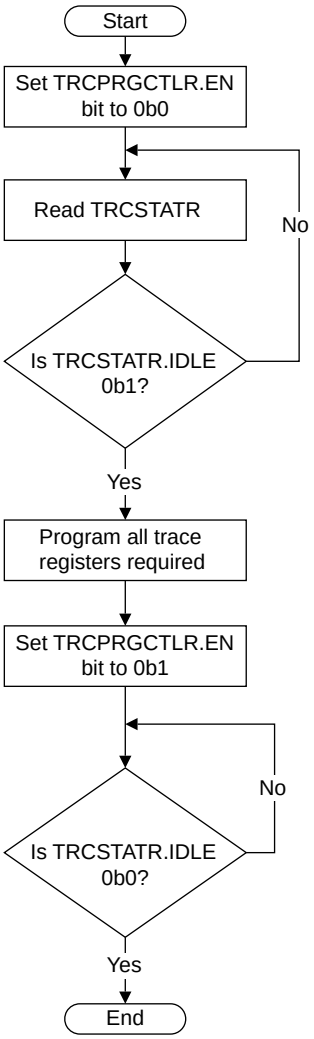
For example, if the counter is reprogrammed, it might start to count based on incorrect events, before the trigger condition has been correctly set up.

To disable instruction trace operations during programming, use:

- The trace program enable bit in the Programming Control Register, TRCPRGCTLR.
- The Status Register, TRCSTATR, to indicate the ETM-M85 status.

The following figure shows the procedure to use.

Figure 4-1: Programming ETM registers



The Cortex®-M85 processor does not have to be in debug state while you program the ETM-M85 registers.

To access the ETM registers, use the external *Advanced Peripheral Bus* (APB) interface. This provides a direct method of programming the ETM.

5 ETM-M85 register descriptions

This chapter describes the ETM-M85 registers.

5.1 Register summary

All ETM-M85 registers are 32 bits wide.


Note

Registers not listed in the following table and not described in this document are not implemented. Reading a non-implemented register address returns zero. Writing to a non-implemented register address has no effect.

The following table:

- Lists the ETM-M85 registers in numerical order and describes each register.
- Includes additional information about each register:
  - The base offset address of the register. The base offset of a register is always four times its register number. For information on the base address of the registers, see *Arm®v8-M Architecture Reference Manual*.
  - The register access type.
  - Additional information about the implementation of the register, where appropriate.

Table 5-1: ETM-M85 register summary

Register number	Base offset	Name	Type	Reset value	Description
1	0x004	TRCPRGCTLR	RW	0x00000000	5.2 TRCPRGCTLR, Programming Control Register on page 36
3	0x00C	TRCSTATR	RO	UNKNOWN	5.3 TRCSTATR, Status Register on page 37
4	0x010	TRCCONFIGR	RW	UNKNOWN	5.4 TRCCONFIGR, Trace Configuration Register on page 38
6	0x018	TRCAUXCTLR	RW	0x00000000	Auxiliary Control Register.
8	0x020	TRCEVENTCTL0R	RW	UNKNOWN	5.5 TRCEVENTCTL0R, Event Control 0 Register on page 39
9	0x024	TRCEVENTCTL1R	RW	UNKNOWN	5.6 TRCEVENTCTL1R, Event Control 1 Register on page 40
11	0x02C	TRCSTALLCTLR	RW	UNKNOWN	5.7 TRCSTALLCTLR, Stall Control Register on page 41
12	0x030	TRCTSCTLR	RW	UNKNOWN	5.8 TRCTSCTLR, Global Timestamp Control Register on page 42
13	0x034	TRCSYNCPR	RO	0x0000000A	5.9 TRCSYNCPR, Synchronization Period Register on page 43
14	0x038	TRCCCCTLR	RW	UNKNOWN	5.10 TRCCCCTLR, Cycle Count Control Register on page 44
16	0x040	TRCTRACEIDR	RW	UNKNOWN	5.11 TRCTRACEIDR, Trace ID Register on page 44
32	0x080	TRCVICTLR	RW	UNKNOWN	5.12 TRCVICTLR, ViewInst Main Control Register on page 45
35	0x08C	TRCVIPCSSCTLR	RW	UNKNOWN	5.12 TRCVICTLR, ViewInst Main Control Register on page 45
72	0x120	TRCEXTINSELR	RW	UNKNOWN	5.14 TRCEXTINSELR, External Input Select Register on page 48
80	0x140	TRCCNTRLDVR0	RW	UNKNOWN	5.15 TRCCNTRLDVR0, Counter Reload Value Register 0 on page 49

Register number	Base offset	Name	Type	Reset value	Description
96	0x180	TRCIDR8	RO	0x00000000	5.16.9 TRCIDR8, ID Register 8 on page 58
97	0x184	TRCIDR9	RO	0x00000000	5.16.10 TRCIDR9, ID Register 9 on page 59
98	0x188	TRCIDR10	RO	0x00000000	5.16.11 TRCIDR10, ID Register 10 on page 59
99	0x18C	TRCIDR11	RO	0x00000000	5.16.12 TRCIDR11, ID Register 11 on page 60
100	0x190	TRCIDR12	RO	0x00000001	5.16.13 TRCIDR12, ID Register 12 on page 61
101	0x194	TRCIDR13	RO	0x00000000	5.16.14 TRCIDR13, ID Register 13 on page 61
112	0x1C0	TRCIMSPEC0	RW	0x00000000	Implementation specific Register 0. There are not **IMPLEMENTATION SPECIFIC** extensions supported, and this register is not implemented.
120	0x1E0	TRCIDR0	RO	0x280006E1	5.16.1 TRCIDR0, ID Register 0 on page 50
121	0x1E4	TRCIDR1	RO	0x4100F450	5.16.2 TRCIDR1, ID Register 1 on page 52
122	0x1E8	TRCIDR2	RO	0x00000004	5.16.3 TRCIDR2, ID Register 2 on page 53
123	0x1EC	TRCIDR3	RO	0x0F090004	5.16.4 TRCIDR3, ID Register 3 on page 54
124	0x1F0	TRCIDR4	RO	-	5.16.5 TRCIDR4, ID Register 4 on page 55
125	0x1F4	TRCIDR5	RO	0x90C70004	5.16.6 TRCIDR5, ID Register 5 on page 57
126	0x1F8	TRCIDR6	-	**UNKNOWN**	Reserved, **RES0**
127	0x1FC	TRCIDR7	-	**UNKNOWN**	Reserved, **RES0**
130-131	0x208-0x20C	TRCRSCTLR2-3	RW	**UNKNOWN**	5.17 TRCRSCTLRn, Resource Selection Registers 2-3 on page 62
160	0x280	TRCSSCCR0	RW	**UNKNOWN**	5.18 TRCSSCCR0, Single-shot Comparator Control Register 0 on page 63
168	0x2A0	TRCSSCSR0	RW	**UNKNOWN**	5.19 TRCSSCSR0, Single-shot Comparator Status Register 0 on page 64
176	0x2C0	TRCSSPCICR0	RW	**UNKNOWN**	5.20 TRCSSPCICR0, Single-shot Processor Comparator Input Control Register 0 on page 65
196	0x310	TRCPDCR	RW	0x00000000	5.21 TRCPDCR, Power Down Control Register on page 66
197	0x314	TRCPDSR	RO	0x00000003	5.22 TRCPDSR, Power Down Status Register on page 67
953	0xEE4	TRCITATBIDR	WO	**UNKNOWN**	5.23.1 TRCITATBIDR, Integration ATB Identification Register on page 69
955	0xEEC	TRCITIDATAR	WO	**UNKNOWN**	5.23.2 TRCITIDATAR, Integration Data Register on page 69
957	0xEF4	TRCITIATBINR	RO	**UNKNOWN**	5.23.3 TRCITIATBINR, Integration Instruction ATB In Register on page 70
959	0xEFC	TRCITIATBOUTR	WO	**UNKNOWN**	5.23.4 TRCITIATBOUTR, Integration Instruction ATB Out Register on page 71
960	0xF00	TRCITCTRL	RW	0x00000000	5.31 TRCCIDR0-3, Component Identification Registers on page 79 5.23.5 TRCITCTRL, Integration Mode Control Register on page 72
1000	0xFA0	TRCCLAIMSET	RW	0x0000000F	5.24 TRCCLAIMSET, Claim Tag Set Register on page 73
1001	0xFA4	TRCCLAIMCLR	RW	0x00000000	5.25 TRCCLAIMCLR, Claim Tag Clear Register on page 74
1006	0xFB8	TRCAUTHSTATUS	RO	**UNKNOWN**	5.26 TRCAUTHSTATUS, Authentication Status Register on page 75
1007	0xFBC	TRCDEVARCH	RO	0x47754A13	5.27 TRCDEVARCH, Device Architecture Register on page 76

[1] **0x00118000** For 8 comparator configuration
0x00114000 For 4 comparator configuration

Register number	Base offset	Name	Type	Reset value	Description
1010	0xFC8	TRCDEVID	RO	0x00000000	5.28 TRCDEVID, Device ID Register on page 76
1011	0xFCC	TRCDEVTYPE	RO	0x00000013	5.29 TRCDEVTYPE, Device Type Register on page 77
1012-1019	0xFD0-0xFEC	TRCPIDR0-7	RO	-	5.30 TRCPIDR0-7, Peripheral Identification Registers on page 77
1020-1023	0xFF0-0xFFC	TRCCIDR0-3	RO	-	5.31 TRCCIDR0-3, Component Identification Registers on page 79

Note

In Table 5-1: ETM-M85 register summary on page 34:

- The Reset value column shows the value of the register immediately after an ETM-M85 reset. For read only registers, every read of the register returns this value.
- Access type is described as follows:

RW	Read and write.
RO	Read only.
WO	Write only.

5.2 TRCPRGCTLR, Programming Control Register

The TRCPRGCTLR enables ETM-M85.

Usage constraints

See 4.1.1 Controlling ETM-M85 programming on page 32.

Configurations

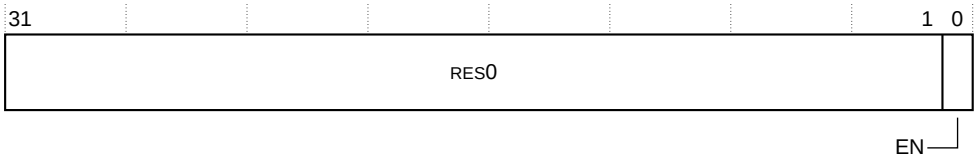
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCPRGCTLR bit assignments.

Figure 5-1: TRCPRGCTLR bit assignments



The following table shows the TRCPRGCTLR bit assignments.

Table 5-2: TRCPRGCTLR bit assignments

Bits	Name	Function
[31:1]	-	**RES0**.
[0]	EN	ETM-M85 enable bit: **0** ETM-M85 is disabled. **1** ETM-M85 is enabled.

5.3 TRCSTATR, Status Register

The TRCSTATR indicates the ETM-M85 status.

Usage constraints

There are no usage constraints.

Configurations

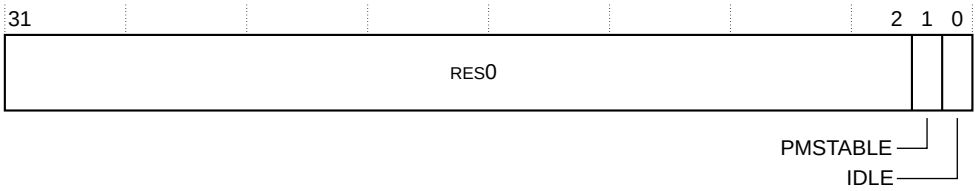
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCSTATR bit assignments.

Figure 5-2: TRCSTATR bit assignments



The following table shows the TRCSTATR bit assignments.

Table 5-3: TRCSTATR bit assignments

Bits	Name	Function
[31:2]	-	**RES0**.
[1]	PMSTABLE	Indicates whether the ETM-M85 registers are stable and can be read: **0** The programmers model is not stable. **1** The programmers model is stable.

Bits	Name	Function
[0]	IDLE	Indicates whether ETM-M85 is inactive: **0** ETM-M85 is not idle. **1** ETM-M85 is idle. When the IDLE bit is set to 1: • ETM-M85 is drained of any trace. • Except for the programming interfaces, all external interfaces on ETM-M85 are quiescent.

5.4 TRCCONFIGR, Trace Configuration Register

The TRCCONFIGR sets the basic tracing options for the ETM-M85.

Usage constraints

This register must always be programmed as part of ETM-M85 initialization.
Only accepts writes when ETM-M85 is disabled.

Configurations

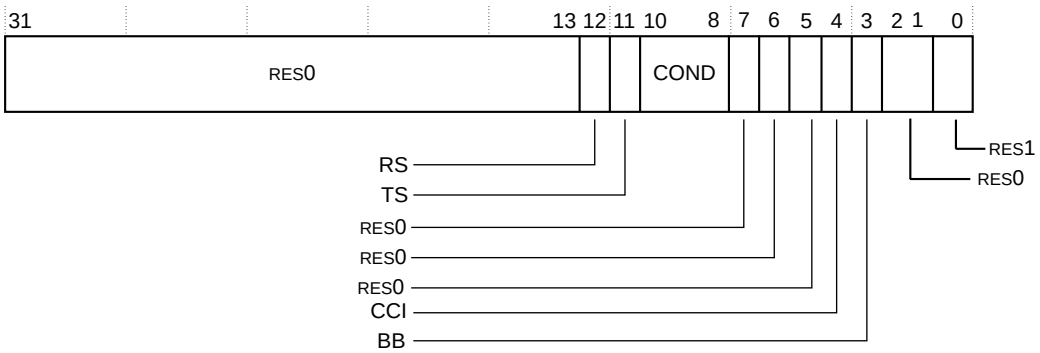
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCCONFIGR bit assignments.

Figure 5-3: TRCCONFIGR bit assignments



The following table shows the TRCCONFIGR bit assignments.

Table 5-4: TRCCONFIGR bit assignments

Bits	Name	Function
[31:13]	-	**RES0**.

Bits	Name	Function
[12]	RS	Return stack enable: **0** Return stack disabled. **1** Return stack enabled.
[11]	TS	Global timestamp tracing: **0** Global timestamp tracing disabled. **1** Global timestamp tracing enabled. For more global timestamp tracing options, see 5.8 TRCTSCTLR, Global Timestamp Control Register on page 42.
[10:8]	COND	Conditional instruction tracing. The supported values are: **0b000** Conditional instruction tracing is disabled. **0b001** Conditional load instructions are traced. **0b010** Conditional store instructions are traced. **0b011** Conditional load and store instructions are traced. **0b111** All conditional instructions are traced. All other values are Reserved.
[7]	VMID	**RES0**.
[6]	CID	**RES0**.
[5]	-	**RES0**.
[4]	CCI	Cycle counting in instruction trace: **0** Cycle counting in instruction trace disabled. **1** Cycle counting in instruction trace enabled. For more cycle counting options, see 5.10 TRCCCCTLR, Cycle Count Control Register on page 44.
[3]	BB	Branch broadcast mode: **0** Branch broadcast mode disabled. **1** Branch broadcast mode enabled.
[2:1]	INSTP0	**RES0**
[0]	-	**RES1**

5.5 TRCEVENTCTL0R, Event Control 0 Register

The TRCEVENTCTL0R controls the tracing of arbitrary events. The events also drive the ETM-M85 external outputs.

Usage constraints

This register must always be programmed as part of ETM-M85 initialization.

Only accepts writes when ETM-M85 is disabled.

Configurations

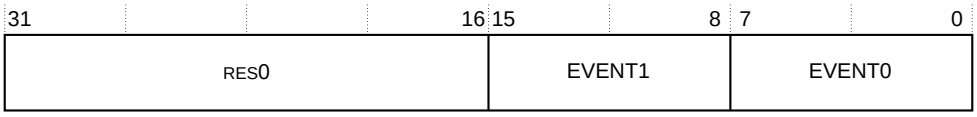
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34 .

The following figure shows the TRCEVENTCTL0R bit assignments.

Figure 5-4: TRCEVENTCTL0R bit assignments



The following table shows the TRCEVENTCTL0R bit assignments.

Table 5-5: TRCEVENTCTL0R bit assignments

Bits	Name	Function
[31:16]	-	**RES0**.
[15:8]	EVENT1	Event selector 1.
[7:0]	EVENT0	Event selector 0.

5.6 TRCEVENTCTL1R, Event Control 1 Register

The TRCEVENTCTL1R controls the events selected by TRCEVENTCTL0R.

See 5.5 TRCEVENTCTL0R, Event Control 0 Register on page 39.

Usage constraints

This register must always be programmed as part of ETM-M85 initialization.

Only accepts writes when ETM-M85 is disabled.

Configurations

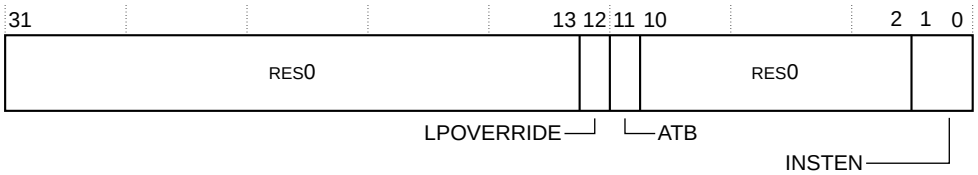
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCEVENTCTL1R bit assignments.

Figure 5-5: TRCEVENTCTL1R bit assignments



The following table shows the TRCEVENTCTL1R bit assignments.

Table 5-6: TRCEVENTCTL1R bit assignments

Bits	Name	Function
[31:13]	-	**RES0.**
[12]	LPOVERRIDE	Low power state behavior override: **0** Low power state behavior unaffected. **1** Low power state behavior overridden. The resources and event trace generation are unaffected by entry to a low power state.
[11]	ATB	ATB trigger enable: **0** ATB trigger disabled. **1** ATB trigger enabled.
[10:2]	-	**RES0**
[1:0]	INSTEN	One bit per event, to enable generation of an event element in the instruction trace stream when the selected event occurs: **0** ETM-M85 does not generate an event element. **1** ETM-M85 generates an event element.

5.7 TRCSTALLCTLR, Stall Control Register

The TRCSTALLCTLR enables ETM-M85 to stall the processor to minimize the risk of overflow if the ETM-M85 FIFO goes over the programmed level.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

This register must always be programmed as part of ETM-M85 initialization.

Configurations

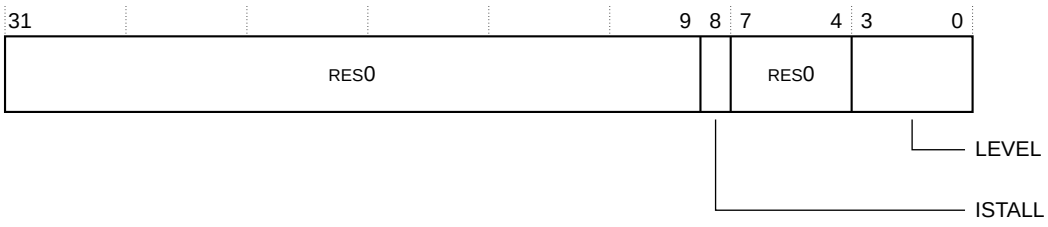
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCSTALLCTLR bit assignments.

Figure 5-6: TRCSTALLCTLR bit assignments



The following table shows the TRCSTALLCTLR bit assignments.

Table 5-7: TRCSTALLCTLR bit assignments

Bits	Name	Function
[31:9]	-	**RES0**.
[8]	ISTALL	Stall processor based on instruction trace buffer space: **0** ETM-M85 must not stall the processor. **1** ETM-M85 can stall the processor. The LEVEL field controls the threshold at which the processor is stalled.
[7:4]	-	**RES0**.
[3:0]	LEVEL	Threshold at which stalling becomes active. This provides four levels. This level can be varied to optimize the level of invasion caused by stalling, balanced against the risk of a FIFO overflow: **0b0000** Zero invasion. This setting has a greater risk of a FIFO overflow. **0b0100** First level of invasion. **0b1000** Second level of invasion. **0b1100** Maximum invasion occurs, but there is less risk of a FIFO overflow. **Note:** Writes to bits[1:0] are ignored and these bits are always set to `0b00`. When the value of this field is `0b0100` or higher, then ETM-M85 might suppress the generation of: • Periodic synchronization in the instruction trace stream. • Global timestamps in the instruction trace stream. • Cycle counting in the instruction trace stream, although the cumulative cycle count remains correct.

5.8 TRCTSCTLR, Global Timestamp Control Register

The TRCTSCTLR controls the insertion of global timestamps into the trace stream. A timestamp is always inserted into the instruction trace stream.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

This register must always be programmed as part of ETM-M85 initialization.

Configurations

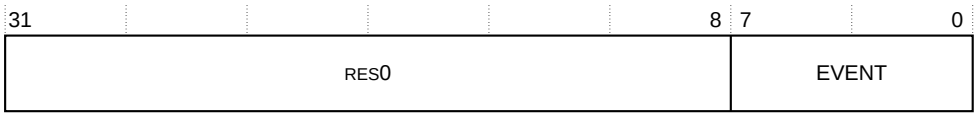
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCTSCTLR bit assignments.

Figure 5-7: TRCTSCTLR bit assignments



The following table shows the TRCTSCTLR bit assignments.

Table 5-8: TRCTSCTLR bit assignments

Bits	Name	Function
[31:8]	-	**RES0**
[7:0]	EVENT	An event selector. When the selected event is triggered, ETM-M85 inserts a global timestamp into the trace streams. For more information on the trace unit event that is selected, see the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*. For more information on resource selection and event selectors, see 3.3.6 Resource selection on page 29.

5.9 TRCSYNCPR, Synchronization Period Register

TRCSYNCPR defines the number of bytes of trace between requests for trace synchronization. This specifies the period of trace synchronization of the trace streams.

Usage constraints

The register is implemented as RO and the synchronization period, which is indicated by the PERIOD bit field, is `0b01010`.

Configurations

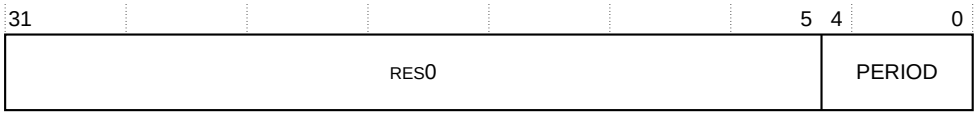
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCSYNCPR bit assignments.

Figure 5-8: TRCSYNCPR bit assignments



The following table shows the TRCSYNCPR bit assignments.

Table 5-9: TRCSYNCPR bit assignments

Bits	Name	Function
[31:5]	-	**RES0**.
[4:0]	PERIOD	Defines the number of bytes of trace between trace synchronization requests as a total of the number of bytes generated by the instruction stream. For ETM-M85, this field is `0b01010`.

5.10 TRCCCCTLR, Cycle Count Control Register

The TRCCCCTLR sets the threshold value for instruction trace cycle counting. The threshold represents the minimum interval between cycle count trace packets.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

This register must always be programmed as part of ETM-M85 initialization.

Configurations

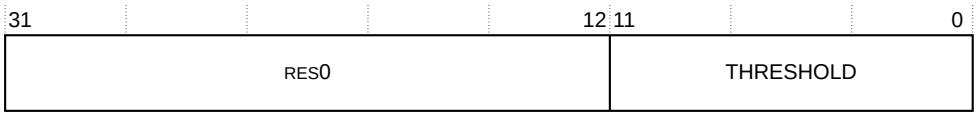
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCCCCTLR bit assignments.

Figure 5-9: TRCCCCTLR bit assignments



The following table shows the TRCCCCTLR bit assignments.

Table 5-10: TRCCCCTLR bit assignments

Bits	Name	Function
[31:12]	-	**RES0**
[11:0]	THRESHOLD	Instruction trace cycle count threshold. If the value for this field is less than `0x4`, then the behavior is **CONSTRAINED UNPREDICTABLE**.

5.11 TRCTRACEIDR, Trace ID Register

The TRCTRACEIDR sets the trace ID on the trace bus. Each trace source in a CoreSight™ system must be programmed with a unique non-reserved trace ID value so trace tools can distinguish trace from different sources.

Usage constraints

In a CoreSight™ system, writing of reserved trace ID values, `0x00` and `0x70-0x7F`, is **UNPREDICTABLE**. This register must always be programmed as part of ETM-M85 initialization.

Only accepts writes when ETM-M85 is disabled.

Configurations

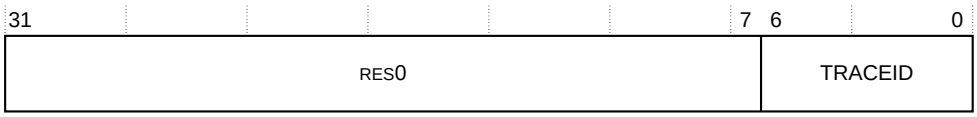
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCTRACEIDR bit assignments.

Figure 5-10: TRCTRACEIDR bit assignments



The following table shows the TRCTRACEIDR bit assignments.

Table 5-11: TRCTRACEIDR bit assignments

Bits	Name	Function
[31:7]	-	**RES0**
[6:0]	TRACEID	Trace ID value. This provides the instruction trace ID.

5.12 TRCVICTLR, ViewInst Main Control Register

The TRCVICTLR controls instruction trace filtering.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

Only returns stable data when TRCSTATR.PMSTABLE is 1.

Must be programmed, particularly to set the value of the SSSTATUS bit, that sets the state of the start-stop logic.

Configurations

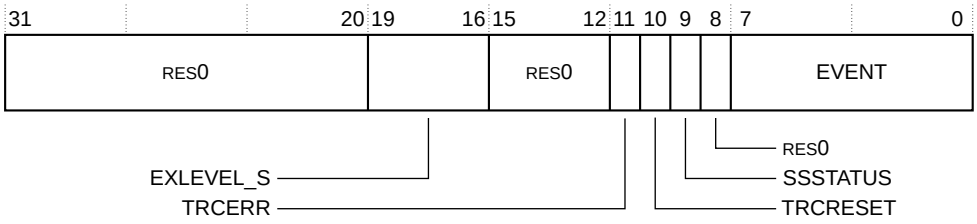
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCVICTLR bit assignments.

Figure 5-11: TRCVICTLR bit assignments



The following table shows the TRCVICTLR bit assignments.

Table 5-12: TRCVICTLR bit assignments

Bits	Name	Function
[31:20]	-	**RES0**
[19:16]	EXLEVEL_S	In Secure state, each bit controls whether instruction tracing is enabled for the corresponding exception level: **0** ETM-M85 does not generate instruction trace, in Secure state, for exception level n. **1** ETM-M85 generates instruction trace, in Secure state, for exception level n. The exception levels are: **Bit[16]** Thread mode. **Bit[17]** **RES0**. **Bit[18]** **RES0**. EXLEVEL_S[2] is never implemented. **Bit[19]** Handler mode.
[15:12]	-	**RES0**.
[11]	TRCERR	Selects whether a system error exception must always be traced: **0** System error exception is traced only if the instruction or exception immediately before the system error exception is traced. **1** System error exception is always traced, regardless of the value of ViewInst.
[10]	TRCRESET	Selects whether a reset exception must always be traced: **0** Reset exception is traced only if the instruction or exception immediately before the reset exception is traced. **1** Reset exception is always traced regardless of the value of ViewInst.
[9]	SSSTATUS	Indicates the current status of the start/stop logic: **0** Start/stop logic is in the stopped state. **1** Start/stop logic is in the started state.
[8]	-	**RES0**

Bits	Name	Function
[7:0]	EVENT	An event selector. For more information on the trace unit event that is selected, see the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4*. For more information on resource selection and event selectors, see 3.3.6 Resource selection on page 29.

5.13 TRCVIPCSSCTLR, ViewInst Start/Stop Processor Comparator Control Register

The TRCVIPCSSCTLR is used to set or read which processor comparator inputs can control the ViewInst start/stop logic.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

This register must be programmed.

Configurations

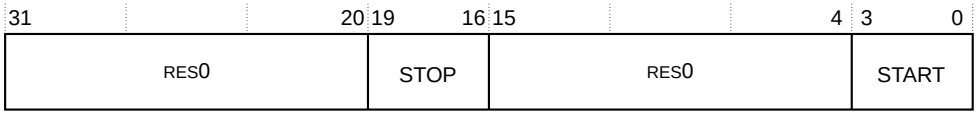
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCVIPCSSCTLR bit assignments when there are four processor comparator inputs implemented for the *Data Watchpoint Trigger* (DWT). TRCIDR4.NUMPC is `0b0100`. For more information on TRCIDR4, see 5.16.5 TRCIDR4, ID Register 4 on page 55.

Figure 5-12: TRCVIPCSSCTLR bit assignments for four processor comparator inputs



The following table shows the TRCVIPCSSCTLR bit assignments for four processor comparator inputs.

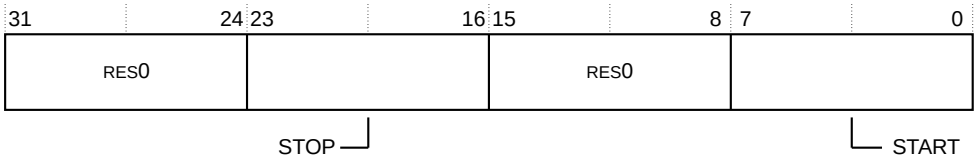
Table 5-13: TRCVIPCSSCTLR bit assignments for four processor comparator inputs

Bits	Name	Function
[31:20]	-	RES0
[19:16]	STOP	Selects which processor comparator inputs are used with ViewInst start/stop control, for the purpose of stopping trace. Each bit represents a processor comparator input, so bit[m] controls the selection of processor comparator input m-16. If bit[m] is: **0** The single processor comparator input m-16 is not selected as a stop resource. **1** The single processor comparator input m-16 is selected as a stop resource.

Bits	Name	Function
[15:4]	-	**RES0.**
[3:0]	START	Selects which processor comparator inputs are used with ViewInst start/stop control, for the purpose of starting trace. Each bit represents a processor comparator input, so bit[n] controls the selection of processor comparator. If bit[n] is: **0** The single processor comparator input n is not selected as a start resource. **1** The single processor comparator input n is selected as a start resource.

The following figure shows the TRCVICTLR bit assignments when there are eight processor comparator inputs implemented for the *Data Watchpoint Trigger* (DWT). TRCIDR4.NUMPC is `0b1000`. For more information on TRCIDR4, see 5.16.5 TRCIDR4, ID Register 4 on page 55.

Figure 5-13: TRCVIPCSSCTLR bit assignments for eight processor comparator inputs



The following table shows the TRCVIPCSSCTLR bit assignments for eight processor comparator inputs.

Table 5-14: TRCVIPCSSCTLR bit assignments for eight processor comparator inputs

Bits	Name	Function
[31:24]	-	**RES0**
[23:16]	STOP	Selects which processor comparator inputs are used with ViewInst start/stop control, for the purpose of stopping trace. Each bit represents a processor comparator input, so bit[m] controls the selection of processor comparator input m-16. If bit[m] is: **0** The single processor comparator input m-16 is not selected as a stop resource. **1** The single processor comparator input m-16 is selected as a stop resource.
[15:8]	-	**RES0.**
[7:0]	START	Selects which processor comparator inputs are used with ViewInst start/stop control, for the purpose of starting trace. Each bit represents a processor comparator input, so bit[n] controls the selection of processor comparator. If bit[n] is: **0** The single processor comparator input n is not selected as a start resource. **1** The single processor comparator input n is selected as a start resource.

5.14 TRCEXTINSELR, External Input Select Register

The TRCEXTINSELR is used to set or read which external inputs are resources to ETM-M85.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

Configurations

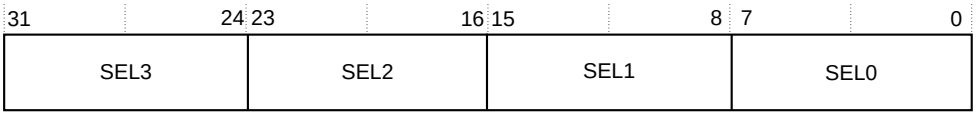
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCEXTINSELR bit assignments.

Figure 5-14: TRCEXTINSELR bit assignments for two processor comparator inputs



The following table shows the TRCEXTINSELR bit assignments.

Table 5-15: TRCEXTINSELR bit assignments

Bits	Name	Function
[31:24]	SEL3	This field is a binary value, of up to 8 bits, that selects which external input is a resource for ETM-M85.
[23:16]	SEL2	This field is a binary value, of up to 8 bits, that selects which external input is a resource for ETM-M85.
[15:8]	SEL1	This field is a binary value, of up to 8 bits, that selects which external input is a resource for ETM-M85.
[7:0]	SEL0	This field is a binary value, of up to 8 bits, that selects which external input is a resource for ETM-M85.

5.15 TRCCNTRLDVR0, Counter Reload Value Register 0

The TRCCNTRLDVR0 register defines the reload value for the reduced function counter.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

The count value is only stable when TRCSTATR.PMSTABLE is 1.

If software uses counter 0, then it must write to this register to set the counter reload value.

Configurations

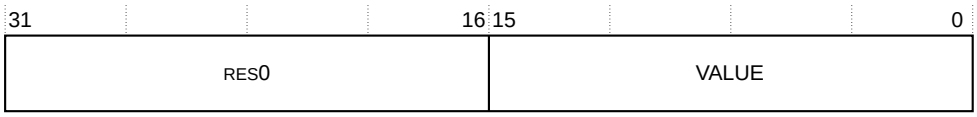
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCCNTRLDVR0 bit assignments.

Figure 5-15: TRCCNTRLDVR0 bit assignments



The following table shows the TRCCNTRLDVR0 bit assignments.

Table 5-16: TRCCNTRLDVR0 bit assignments

Bits	Value	Function
[31:16]	-	**RES0**.
[15:0]	VALUE	Defines the reload value for the counter. This value is loaded into the counter each time the reload event occurs.

5.16 TRCIDR0-13, ID Registers

This section describes the ETM-M85 ID registers.

5.16.1 TRCIDR0, ID Register 0

The TRCIDR0 indicates the tracing capabilities of the ETM-M85 instruction trace.

Usage constraints

This register is read-only.

Configurations

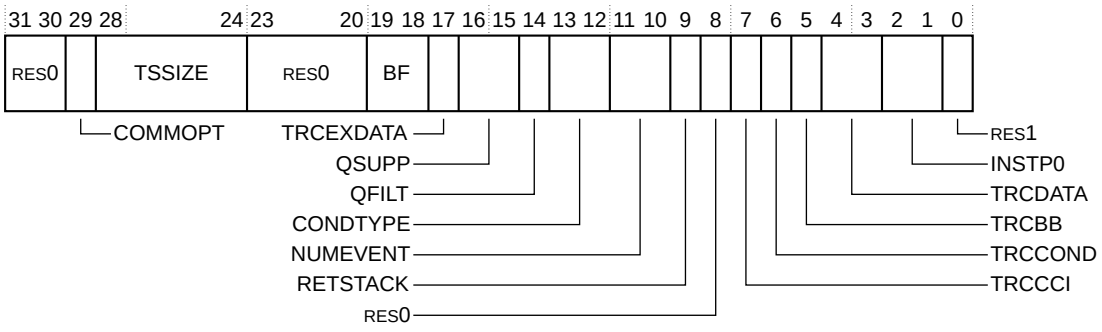
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR0 bit assignments.

Figure 5-16: TRCIDR0 bit assignments



The following table shows the TRCIDR0 bit assignments.

Table 5-17: TRCIDR0 bit assignments

Bits	Name	Function
[31:30]	-	**RES0.**
[29]	COMMOPT	Indicates the meaning of the commit field in some packets: **1** Commit mode 1.
[28:24]	TSSIZE	Global timestamp size: **0b01000** Maximum of 64-bit global timestamp implemented.
[23:20]	-	Reserved, **RES0.**
[19:18]	BF	Branch Future Support. **0b00** Branch future not supported.
[17]	TRCEXDATA	Indicates support for the tracing of data transfers for exceptions and exception returns: **0** TRCVDCTLR.TRCEXDATA is not implemented.
[16:15]	QSUPP	Indicates Q element support: **0b00** Q elements not supported.
[14]	QFILT	**RES0.**
[13:12]	CONDTYPE	Indicates how conditional results are traced: **0b00** ETM-M85 indicates only if a conditional instruction passes or fails its condition code check.
[11:10]	NUMEVENT	Number of events supported in the trace: **0b01** Two events supported.
[9]	RETSTACK	Return stack support: **1** Two entry return stack implemented.
[8]	-	**RES0.**

Bits	Name	Function
[7]	TRCCCI	Support for cycle counting in the instruction trace: **1** Cycle counting in the instruction trace is implemented.
[6]	TRCCOND	Support for conditional instruction tracing: **1** Conditional instruction tracing is implemented.
[5]	TRCBB	Support for branch broadcast tracing: **1** Branch broadcast tracing is implemented.
[4:3]	TRCDATA	Support for tracing of data: **0b00** Data tracing is not supported.
[2:1]	INSTP0	Support for tracing of load and store instructions as P0 elements: **0b00** Tracing of load and store instructions as P0 elements is not supported.
[0]	-	**RES1**

5.16.2 TRCIDR1, ID Register 1

The TRCIDR1 indicates the ETM-M85 architecture.

Usage constraints

This register is read-only.

Configurations

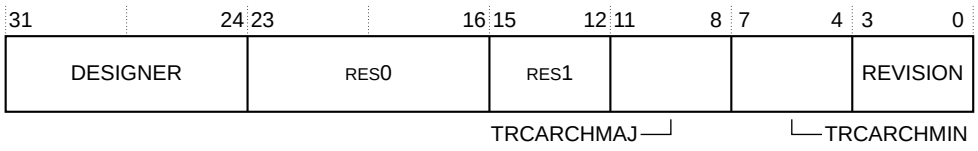
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR1 bit assignments.

Figure 5-17: TRCIDR1 bit assignments



The following table shows the TRCIDR1 bit assignments.

Table 5-18: TRCIDR1 bit assignments

Bits	Name	Function
[31:24]	DESIGNER	Indicates the designer of the trace unit: **0x41** Arm

Bits	Name	Function
[23:16]	-	**RES0**.
[15:12]	-	**RES1**.
[11:8]	TRCARCHMAJ	Major ETM-M85 architecture version number: **0x4** ETMv4.
[7:4]	TRCARCHMIN	Minor ETM-M85 architecture version number: **0x5** Minor revision 5.
[3:0]	REVISION	Implementation revision number: **0x1** Implementation revision `0x1`.

5.16.3 TRCIDR2, ID Register 2

The TRCIDR2 indicates the maximum sizes of certain aspects of items in the trace.

Usage constraints

This register is read-only.

Configurations

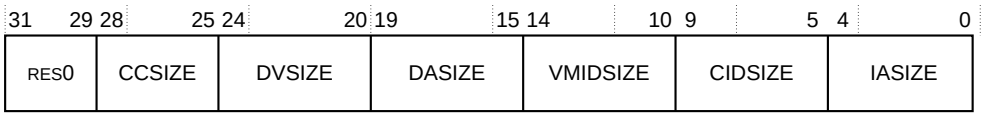
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR2 bit assignments.

Figure 5-18: TRCIDR2 bit assignments



The following table shows the TRCIDR2 bit assignments.

Table 5-19: TRCIDR2 bit assignments

Bits	Name	Function
[31:29]	-	**RES0**.
[28:25]	CCSIZE	Indicates the size of the cycle counter in bits minus 12: **0b0000** Cycle count is 12 bits in length.
[24:20]	DVSIZE	Data value size in bytes: **0b00000** Data value size not supported.

Bits	Name	Function
[19:15]	DASIZE	Data address size in bytes: **0b00000** Data address size not supported.
[14:10]	VMIDSIZE	Virtual Machine ID size: **0b00000** Virtual Machine ID tracing not implemented.
[9:5]	CIDSIZE	Context ID tracing: **0b00000** Context ID tracing not implemented.
[4:0]	IASIZE	Instruction address size: **0b00100** Maximum of 32-bit address size.

5.16.4 TRCIDR3, ID Register 3

The TRCIDR3 indicates certain aspects of the ETM-M85 configuration.

Usage constraints

This register is read-only.

Configurations

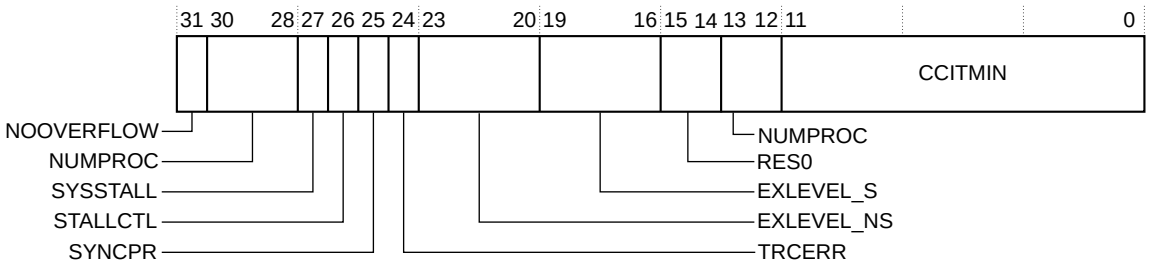
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR3 bit assignments.

Figure 5-19: TRCIDR3 bit assignments



The following table shows the TRCIDR3 bit assignments.

Table 5-20: TRCIDR3 bit assignments

Bits	Name	Function
[31]	NOOVERFLOW	Indicates whether TRCSTALLCTLR.NOOVERFLOW is implemented: **0** NOOVERFLOW is not implemented.
[30:28]	NUMPROC	Indicates the number of processors available for tracing. **0b000** ETM-M85 can trace one processor. NUMPROC uses bits [30:28] and bits [13:12] to form a single 5-bit field.Bits [13:12] form the top bits of this field.
[27]	SYSSTALL	System support for stall control of the processor. **1** System supports stall control of the processor. This field is used with STALLCTL. The system supports stalling of the processor only when both SYSSTALL and STALLCTL are 1 .
[26]	STALLCTL	Stall control support: **1** TRCSTALLCTLR is implemented.
[25]	SYNCPR	Indicates trace synchronization period support: **1** TRCSYNCPR is read-only for instruction trace only configuration. The trace synchronization period is fixed.
[24]	TRCERR	Indicates whether TRCVICTLR.TRCERR is implemented: **1** TRCERR is implemented.
[23:20]	EXLEVEL_NS	**RES0**
[19:16]	EXLEVEL_S	Exception levels implemented. One bit for each level. **0b1001** Privilege levels Thread and Handler are implemented. Thread is at exception level 0 and Handler is at exception level 3.
[15:14]	-	**RES0.**
[13:12]	NUMPROC	Indicates the number of processors available for tracing. **0b00** ETM-M85 can trace one processor. NUMPROC uses bits [30:28] and bits [13:12] to form a single 5-bit field.Bits [13:12] form the top bits of this field.
[11:0]	CCITMIN	Minimum value which can be programmed to TRCCCCTLR.THRESHOLD, defining the minimum cycle counting threshold. **0x4** Minimum of four instruction trace cycles.

5.16.5 TRCIDR4, ID Register 4

The TRCIDR4 indicates the available ETM-M85 resources.

Usage constraints

This register is read-only

Configurations

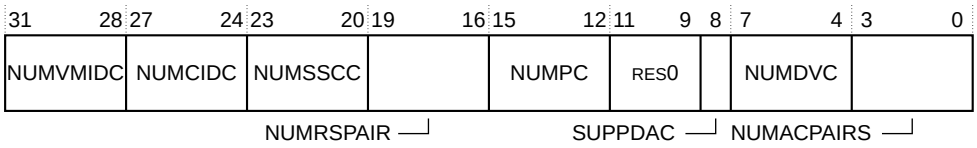
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR4 bit assignments.

Figure 5-20: TRCIDR4 bit assignments



The following table shows the TRCIDR4 bit assignments.

Table 5-21: TRCIDR4 bit assignments

Bits	Name	Function
[31:28]	NUMVMIDC	Number of *Virtual Machine ID* (VMID) comparators implemented: **0b0000** VMID comparators are not implemented.
[27:24]	NUMCIDC	Number of Context ID comparators implemented: **0b0000** Context ID comparators are not supported.
[23:20]	NUMSSCC	Number of single-shot comparator controls implemented: **0b0001** One single-shot comparator control is implemented.
[19:16]	NUMRSPAIR	Number of resource selection pairs implemented: **0b0001** Two resource selection pairs are implemented.
[15:12]	NUMPC	Number of processor comparator inputs implemented for the DWT: **0b0100** Four processor comparator inputs. **0b1000** Eight processor comparator inputs.
[11:9]	-	**RES0**
[8]	SUPPDAC	Data address comparisons implemented: **0** Data address comparisons are not supported.

Bits	Name	Function
[7:4]	NUMDVC	Number of data value comparators implemented: **0b0000** No data value comparators are implemented.
[3:0]	NUMACPAIRS	Number of address comparator pairs implemented: **0b0000** No address comparator pairs are implemented.

5.16.6 TRCIDR5, ID Register 5

The TRCIDR5 indicates the available ETM-M85 resources.

Usage constraints

This register is read-only.

Configurations

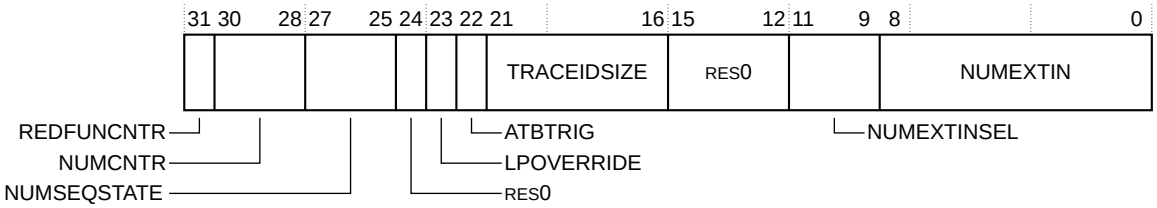
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR5 bit assignments.

Figure 5-21: TRCIDR5 bit assignments



The following table shows the TRCIDR5 bit assignments.

Table 5-22: TRCIDR5 bit assignments

Bits	Name	Function
[31]	REDFUNCNTR	Reduced Function Counter implemented: **1** Counter 0 is implemented as a Reduced Function Counter.
[30:28]	NUMCNTR	Number of counters implemented: **0b001** One counter implemented.
[27:25]	NUMSEQSTATE	Number of sequencer states implemented: **0b000** No sequencer states implemented.
[24]	-	**RES0**

Bits	Name	Function
[23]	LPOVERRIDE	Low-power state override support: **1** Low-power state override support implemented.
[22]	ATBTRIG	ATB trigger support: **1** ATB trigger support implemented.
[21:16]	TRACEIDSIZE	Number of bits of trace ID: **0x07** 7-bit trace ID implemented.
[15:12]	-	**RES0**.
[11:9]	NUMEXTINSEL	Number of input selectors implemented. **0b100** Four input selectors, SEL0 to SEL3, are present.
[8:0]	NUMEXTIN	Number of external inputs implemented: **0x4+ number of *Performance Monitor Unit* (PMU) events** Four external inputs and any PMU event inputs are implemented.

5.16.7 TRCIDR6, ID Register 6

The TRCIDR6 is reserved, **RES0**.

5.16.8 TRCIDR7, ID Register 7

The TRCIDR7 is reserved, **RES0**.

5.16.9 TRCIDR8, ID Register 8

The TRCIDR8 indicates the maximum speculation depth of the instruction trace stream.

Usage constraints

This register is read-only.

Configurations

This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR8 bit assignments.

Figure 5-22: TRCIDR8 bit assignments

The following table shows the TRCIDR8 bit assignments.

Table 5-23: TRCIDR8 bit assignments

Bits	Name	Function
[31:0]	MAXSPEC	This is the maximum number of P0 elements that have not been committed in the trace stream at any time. **0x00000000** Maximum trace speculation depth is zero.

5.16.10 TRCIDR9, ID Register 9

The TRCIDRn indicates the number of P0 right-hand keys that are used.

Usage constraints

This register is read-only.

Configurations

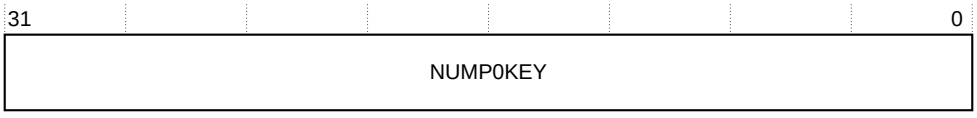
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR9 bit assignments.

Figure 5-23: TRCIDR9 bit assignments



The following table shows the TRCIDR9 bit assignments.

Table 5-24: TRCIDR9 bit assignments

Bits	Name	Function
[31:0]	NUMP0KEY	**0x00000000** No P0 keys used in instruction trace.

5.16.11 TRCIDR10, ID Register 10

The TRCIDR10 indicates the total number of P1 right-hand keys, including normal and special keys.

Usage constraints

This register is read-only.

Configurations

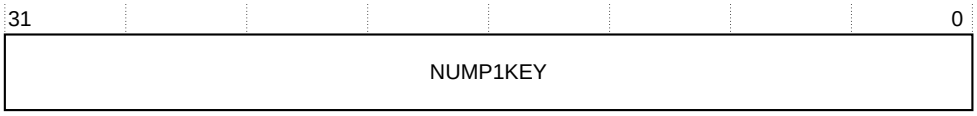
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR10 bit assignments.

Figure 5-24: TRCIDR10 bit assignments



The following table shows the TRCIDR10 bit assignments.

Table 5-25: TRCIDR10 bit assignments

Bits	Name	Function	
[31:0]	NUMP1KEY	**0x00000000**	No P1 right-hand keys used in instruction trace.

5.16.12 TRCIDR11, ID Register 11

The TRCIDR11 indicates the number of special P1 right-hand keys.

Usage constraints

This register is read-only

Configurations

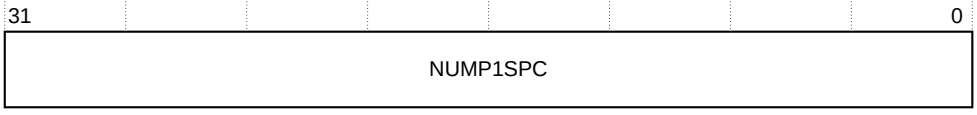
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR11 bit assignments.

Figure 5-25: TRCIDR11 bit assignments



The following table shows the TRCIDR11 bit assignments.

Table 5-26: TRCIDR11 bit assignments

Bits	Name	Function
[31:0]	NUMP1SPC	**0x00000000** No special P1 right-hand keys used in any configuration.

5.16.13 TRCIDR12, ID Register 12

The TRCIDR12 indicates the total number of conditional instruction right-hand keys, including normal and special keys.

Usage constraints

This register is read-only.

Configurations

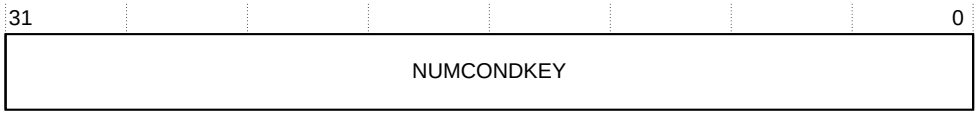
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR12 bit assignments.

Figure 5-26: TRCIDR12 bit assignments



The following table shows the TRCIDR12 bit assignments.

Table 5-27: TRCIDR12 bit assignments

Bits	Name	Function
[31:0]	NUMCONDKEY	**0x00000001** One conditional instruction right-hand key implemented.

5.16.14 TRCIDR13, ID Register 13

The TRCIDR13 indicates the number of special conditional instruction right-hand keys.

Usage constraints

This register is read-only.

Configurations

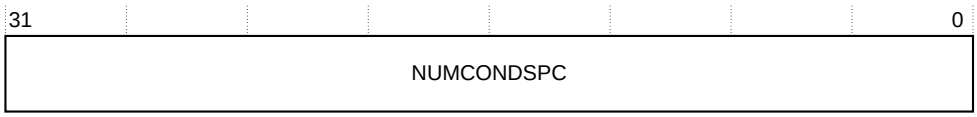
This register is available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCIDR13 bit assignments.

Figure 5-27: TRCIDR13 bit assignments



The following table shows the TRCIDR13 bit assignments.

Table 5-28: TRCIDR13 bit assignments

Bits	Name	Function
[31:0]	NUMCONDSPC	**0x00000000** No special conditional instruction right-hand keys implemented.

5.17 TRCRSCTLRn, Resource Selection Registers 2-3

The TRCRSCTLRn controls the selection of trace resources.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

Configurations

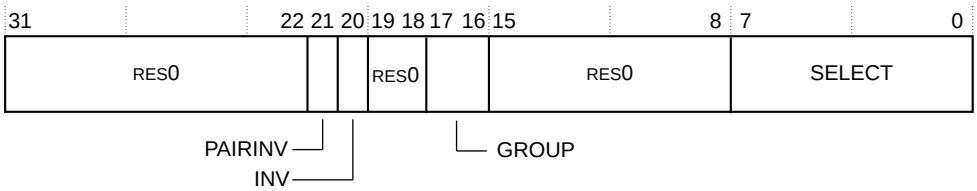
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCRSCTLRn bit assignments.

Figure 5-28: TRCRSCTLRn bit assignments



The following table shows the TRCRSCTLRn bit assignments.

Table 5-29: TRCRSCTLRn bit assignments

Bits	Name	Function
[31:22]	-	**RES0**.
[21]	PAIRINV	Inverts the result of a combined pair of resources. This bit is only implemented on TRCRSCTLR2 and not on TRCRSCTLR3.
[20]	INV	Inverts the selected resources: **0** Resource is not inverted. **1** Resource is inverted.
[19:18]	-	**RES0**.
[17:16]	GROUP	Selects a group of resources.
[15:8]	-	**RES0**.
[7:0]	SELECT	Selects one or more resources from the wanted group. One bit is provided per resource from the group.

The following table lists which resources are selected, depending on the values of the GROUP and SELECT fields.

Table 5-30: Resource selection

Group	Select	Resource
0b0000	Bits 0-3	External input selectors 0-3. When select bit N is set, then the resource selector is sensitive to external input selector N.
0b0001	Bits 0-7	When select bit N is set, the resource selector is sensitive to processor comparator input N.
0b0010	0	Counter at zero 0
0b0011	0	Single-Shot Comparator control 0
0b0100-0b1111	0-15	Reserved

5.18 TRCSSCCR0, Single-shot Comparator Control Register 0

The TRCSSCCR0 controls the single-shot comparator.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

Configurations

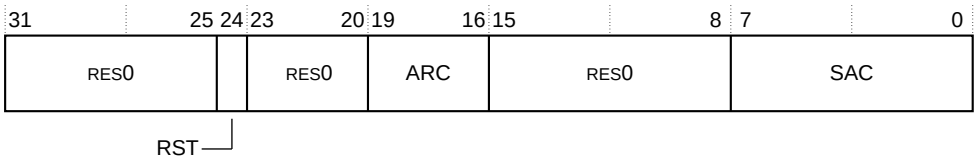
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCSSCCR0 bit assignments.

Figure 5-29: TRCSSCCR0 bit assignments



The following table shows the TRCSSCCR0 bit assignments.

Table 5-31: TRCSSCCR0 bit assignments

Bits	Name	Function
[31:25]	-	**RES0**.
[24]	RST	Enables the single-shot comparator resource to be reset when it occurs, to enable another comparator match to be detected: **0** When the single-shot comparator resource fires, it is not reset. **1** When the single-shot comparator resource fires, it is reset. This enables the single-shot comparator resource to fire multiple times.
[23:20]	-	**RES0**.
[19:16]	ARC	RAZ/WI.
[15:8]	-	**RES0**.
[7:0]	SAC	RAZ/WI.

5.19 TRCSSCSR0, Single-shot Comparator Status Register 0

The TRCSSCSR0 indicates the status of the single-shot comparators. TRCSSCSR0 is sensitive to instruction addresses.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

Configurations

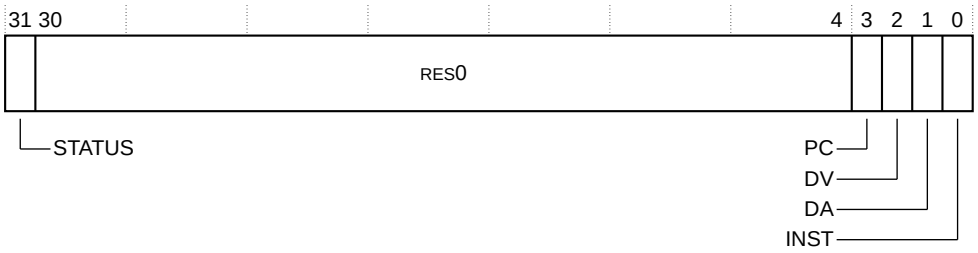
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCSSCSR0 bit assignments.

Figure 5-30: TRCSSCSR0 bit assignments



The following table shows the TRCSSCSR0 bit assignments.

Table 5-32: TRCSSCSR0 bit assignments

Bits	Name	Function
[31]	STATUS	Single-shot status. This indicates whether any of the selected comparators have matched: **0** Match has not occurred. **1** Match has occurred at least once. When programming ETM-M85, if TRCSSCCR0.RST is 0, the STATUS bit must be explicitly written to 0 to enable this single-shot comparator control. If TRCSSCCR0.RST is set, the STATUS bit clears automatically so it is not necessary to clear the STATUS bit when programming. Otherwise, the STATUS bit must be cleared when programming ETM-M85 otherwise a single-shot comparison cannot occur.
[30:4]	-	**RES0**
[3]	PC	Indicates that the Single-shot comparator is sensitive to processor comparator inputs: **1** Single-shot comparator is sensitive to processor comparator inputs.
[2]	DV	Data value comparator support: **0** Single-shot data value comparisons not supported.
[1]	DA	Data address comparator support: **0** Single-shot data address comparisons not supported.
[0]	INST	Instruction address comparator support: **0** Single-shot instruction address comparisons not supported.

5.20 TRCSSPCICR0, Single-shot Processor Comparator Input Control Register 0

The TRCSSPCICR0 selects the processor comparator inputs for Single-shot control.

Usage constraints

Only accepts writes when ETM-M85 is disabled.

Configurations

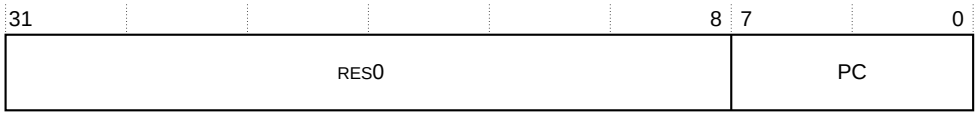
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCSSPCICR0 bit assignments.

Figure 5-31: TRCSSPCICR0 bit assignments



The following table shows the TRCSSPCICR0 bit assignments.

Table 5-33: TRCSSPCICR0 bit assignments

Bits	Name	Function
[31:8]	-	RES0
[7:0]	PC	Selects one or more processor comparator inputs for Single-shot control. One bit is provided for each processor comparator input. The number of comparator inputs can be either four or eight.

5.21 TRCPDCR, Power Down Control Register

The TRCPDCR request to the system power controller to keep ETM-M85 powered up.

Usage constraints

There are no usage constraints.

Configurations

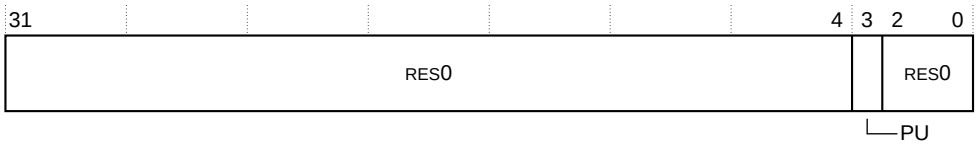
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCPDCR bit assignments.

Figure 5-32: TRCPDCR bit assignments



The following table shows the TRCPDCR bit assignments.

Table 5-34: TRCPDCR bit assignments

Bits	Name	Function
[31:4]	-	**RES0**
[3]	PU	Power up request, to request that power to ETM-M85 and access to the trace registers is maintained: **0** Power not requested. **1** Power requested. This bit is reset to 0 on ETM-M85 reset.
[2:0]	-	**RES0**

5.22 TRCPDSR, Power Down Status Register

The TRCPDSR indicates the power down status of the ETM-M85.

Usage constraints

There are no usage constraints.

Configurations

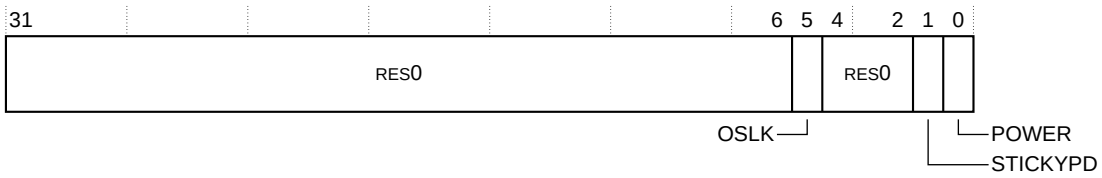
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCPDSR bit assignments.

Figure 5-33: TRCPDSR bit assignments



The following table shows the TRCPDSR bit assignments.

Table 5-35: TRCPDSR bit assignments

Bits	Name	Function
[31:6]	-	**RES0**
[5]	OSLK	**RES0**
[4:2]	-	**RES0**
[1]	STICKYPD	Sticky power down state. **0** Trace register power has not been removed since the TRCPDSR was last read. **1** Trace register power has been removed since the TRCPDSR was last read. This bit is set to 1 when power to the ETM-M85 registers is removed, to indicate that programming state has been lost. It is cleared after a read of the TRCPDSR.
[0]	POWER	Indicates ETM-M85 is powered up: **1** ETM-M85 is powered up. All registers are accessible. If a system implementation allows ETM-M85 to be powered down independently of the debug power domain, the system must ensure: • Accesses to ETM-M85 complete correctly. • Reads to this location return 0 to indicate that ETM-M85 can be powered down.

5.23 Integration test registers

The ETM-M85 integration test registers can be used to access some of the ports that are useful in determining the system level trace topology, by identifying the integration between specific components. Because the integration mode overrides the normal bus protocols, the ETM and ATB interconnect must be reset when any topology detection has been performed. Integration test registers are used to set the outputs and read the state of some of the signals.

To access the integration test registers, you must first set bit[0] of the 5.23.5 TRCITCTRL, Integration Mode Control Register on page 72 to 1.

- You can use the write-only integration test registers to set the outputs of some of the ETM-M85 signals. The following table shows the signals that can be controlled in this way.

Table 5-36: Output signals that the integration test registers can control

Signal	Register	Bits	Register description
AFREADYE	TRCITIATBOUTR	[1]	5.23.4 TRCITIATBOUTR, Integration Instruction ATB Out Register on page 71
ATIDE[6:0]	TRCITATBIDR	[6:0]	5.23.1 TRCITATBIDR, Integration ATB Identification Register on page 69
ATDATAE[6:0]	TRCITIDATAR	[6:0]	5.23.2 TRCITIDATAR, Integration Data Register on page 69
ATVALIDE	TRCITIATBOUTR	[0]	5.23.4 TRCITIATBOUTR, Integration Instruction ATB Out Register on page 71

- You can use the read-only integration test registers to read the state of some of the ETM-M85 input signals. The following table shows the signals that can be read in this way.

Table 5-37: Input signals that the integration test registers can read

Signal	Register	Bits	Register description
AFVALIDE	TRCITIATBINR	[1]	5.23.3 TRCITIATBINR, Integration Instruction ATB In Register on page 70
ATREADYE	TRCITIATBINR	[0]	5.23.3 TRCITIATBINR, Integration Instruction ATB In Register on page 70

See the *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* for more information about TRCITCTRL.

5.23.1 TRCITATBIDR, Integration ATB Identification Register

The TRCITATBIDR sets the state of output pins.

The output pins are listed in Table 5-38: TRCITATBIDR bit assignments on page 69.

Usage constraints

- Available when bit[0] of TRCITCTRL is set to 1.
- The value of the register sets the signals on the output pins when the register is written.
- This is a write-only register.

Configurations

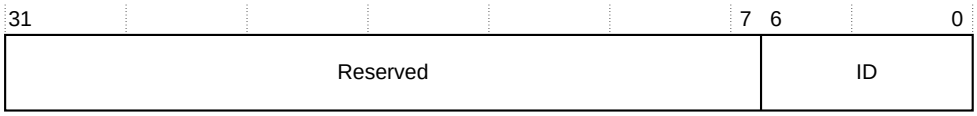
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34 and Table 5-36: Output signals that the integration test registers can control on page 68.

The following figure shows the TRCITATBIDR bit assignments.

Figure 5-34: TRCITATBIDR bit assignments



The following table shows the TRCITATBIDR bit assignments.

Table 5-38: TRCITATBIDR bit assignments

Bits	Name	Function
[31:7]	-	**RES0**.
[6:0]	ID	Drives the **ATIDE[6:0]** output pin.

5.23.2 TRCITIDATAR, Integration Data Register

The TRCITIDATAR sets the state of output pins.

The output pins are listed in Table 5-39: TRCITIDATAR bit assignments on page 70.

Usage constraints

- Available when bit[0] of TRCITCTRL is set to 1.
- The value of the register sets the signals on the output pins when the register is written.
- This is a write-only register.

Configurations

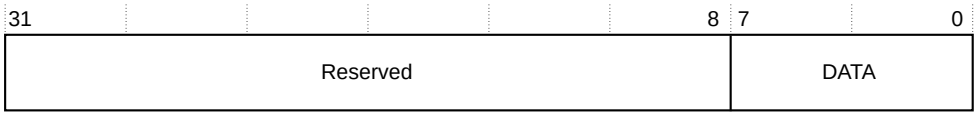
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34 and Table 5-36: Output signals that the integration test registers can control on page 68.

The following figure shows the TRCITIDATAR bit assignments.

Figure 5-35: TRCITIDATAR bit assignments



The following table shows the TRCITIDATAR bit assignments.

Table 5-39: TRCITIDATAR bit assignments

Bits	Name	Function
[31:8]	-	**RES0**.
[7:0]	DATA	Drives the **ATDATAE[7:0]** output pin.

5.23.3 TRCITIATBINR, Integration Instruction ATB In Register

The TRCITIATBINR reads the state of the input pins.

The input pins are listed in Table 5-40: TRCITIATBINR bit assignments on page 71.

Usage constraints

- Available when bit[0] of TRCITCTRL is set to 1.
- The values of the register bits depend on the signals on the input pins when the register is read.

Configurations

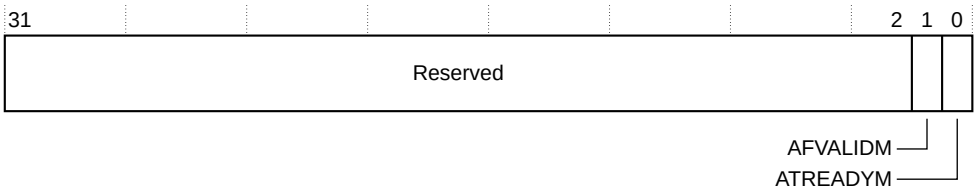
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34 and Table 5-36: Output signals that the integration test registers can control on page 68.

The following figure shows the TRCITIATBINR bit assignments.

Figure 5-36: TRCITIATBINR bit assignments



The following table shows the TRCITIATBINR bit assignments.

Table 5-40: TRCITIATBINR bit assignments

Bits	Name	Function
[31:2]	-	Reserved. Read undefined.
[1]	AFVALIDM	Returns the value of the **AFVALIDE** input pin[2].
[0]	ATREADYM	Returns the value of the **ATREADYE** input pin.

5.23.4 TRCITIATBOUTR, Integration Instruction ATB Out Register

The TRCITIATBOUTR sets the state of the output pins.

These output pins are listed in Table 5-41: TRCITIATBOUTR bit assignments on page 72.

Usage constraints

- Available when bit[0] of TRCITCTRL is set to 1.
- The value of the register sets the signals on the output pins when the register is written.
- This is a write-only register.

Configurations

Available in all configurations.

[2] When an input pin is LOW, the corresponding register bit is 0.

When an input pin is HIGH, the corresponding register bit is 1.

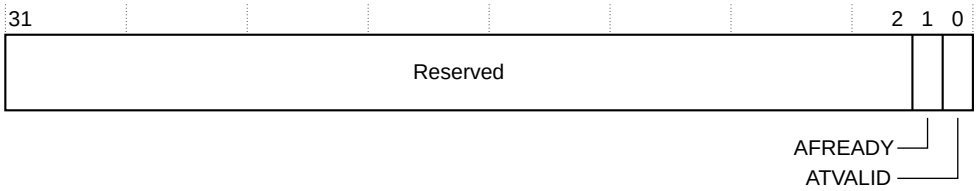
The TRCITIATBINR bit values always correspond to the physical state of the input pins.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34 and Table 5-36: Output signals that the integration test registers can control on page 68.

The following figure shows the TRCITIATBOUTR bit assignments.

Figure 5-37: TRCITIATBOUTR bit assignments



The following table shows the TRCITIATBOUTR bit assignments.

Table 5-41: TRCITIATBOUTR bit assignments

Bits	Name	Function
[31:2]	-	Reserved. Read undefined.
[1]	AFREADY	Drives the **AFREADYE** output pin.
[0]	ATVALID	Drives the **ATVALIDE** output pin.

5.23.5 TRCITCTRL, Integration Mode Control Register

The TRCITCTRL enables topology detection or integration testing, by putting ETM-M85 into integration mode.

Usage constraints

Arm recommends that you perform a debug reset after using integration mode.

Configurations

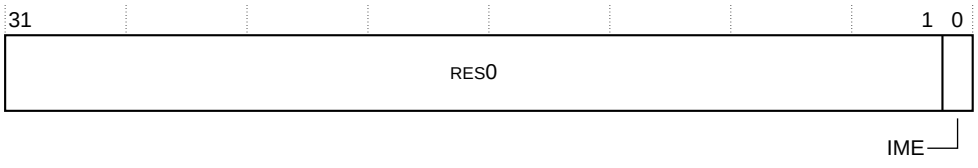
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCITCTRL bit assignments.

Figure 5-38: TRCITCTRL bit assignments



The following table shows the TRCITCTRL bit assignments.

Table 5-42: TRCITCTRL bit assignments

Bits	Name	Function
[31:1]	-	**RES0**
[0]	IME	Integration mode enable: **0** ETM-M85 is not in integration mode. This is the reset value. **1** ETM-M85 is in integration mode.

5.24 TRCCLAIMSET, Claim Tag Set Register

The TRCCLAIMSET sets bits in the claim tag and determines the number of claim tag bits implemented.

Usage constraints

There are no usage constraints.

Configurations

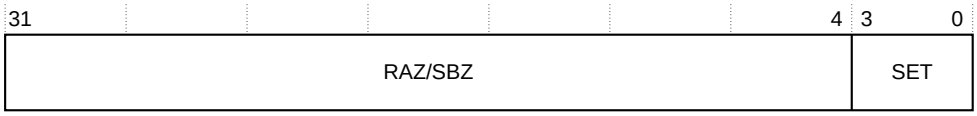
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCCLAIMSET bit assignments.

Figure 5-39: TRCCLAIMSET bit assignments



The following table shows the TRCCLAIMSET bit assignments.

Table 5-43: TRCCLAIMSET bit assignments

Bits	Name	Function
[31:4]	-	RAZ/SBZ.

Bits	Name	Function
[3:0]	SET	On reads, each bit is: **1** Claim tag bit is implemented. This value is returned for each of the claim bits 0-3, indicating 4 claim bits are implemented. On writes, setting each bit to: **0** Has no effect. **1** Sets the relevant bit of the claim tag.

5.25 TRCCLAIMCLR, Claim Tag Clear Register

The TRCCLAIMCLR clears bits in the claim tag and determines the current value of the claim tag.

Usage constraints

There are no usage constraints.

Configurations

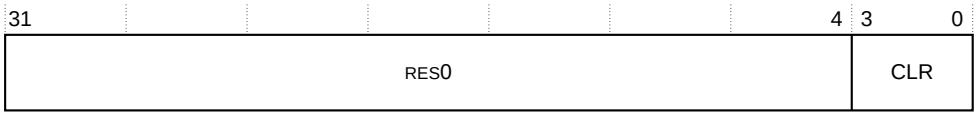
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCCLAIMCLR bit assignments.

Figure 5-40: TRCCLAIMCLR bit assignments



The following table shows the TRCCLAIMCLR bit assignments.

Table 5-44: TRCCLAIMCLR bit assignments

Bits	Name	Function
[31:4]	-	**RES0**.
[3:0]	CLR	On reads, for each bit: **0** Claim tag bit is not set. **1** Claim tag bit is set. On writes, for each bit: **0** Has no effect. **1** Clears the relevant bit of the claim tag.

5.26 TRCAUTHSTATUS, Authentication Status Register

The TRCAUTHSTATUS indicates the current level of tracing permitted by the system.

Usage constraints

There are no usage constraints.

Configurations

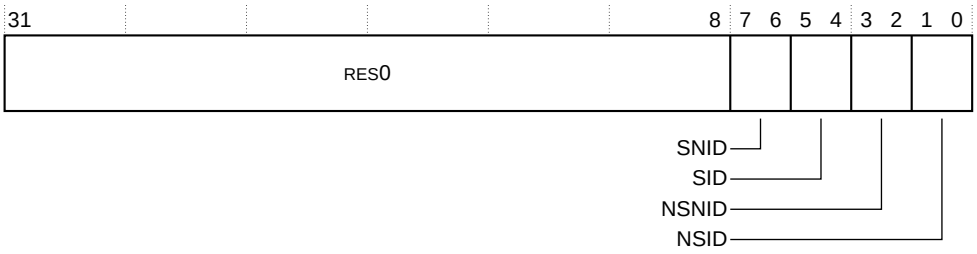
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCAUTHSTATUS bit assignments.

Figure 5-41: TRCAUTHSTATUS bit assignments



The following table shows the TRCAUTHSTATUS bit assignments.

Table 5-45: TRCAUTHSTATUS bit assignments

Bits	Name	Function
[31:8]	-	RES0
[7:6]	SNID[3]	Secure Non-Invasive Debug: **0b00** Secure Non-Invasive Debug not implemented. **0b10** Secure Non-Invasive Debug implemented, but disabled. **0b11** Secure Non-Invasive Debug implemented and enabled.
[5:4]	SID	Secure Invasive Debug: **0b00** Secure Invasive Debug not implemented.
[3:2]	NSNID	Non-secure Non-Invasive Debug: **0b10** Non-secure Non-Invasive Debug implemented, but disabled. **0b11** Non-secure Non-Invasive Debug implemented and enabled.
[1:0]	NSID	Non-secure Invasive Debug: **0b00** Non-secure Invasive Debug not implemented.

[3] SNID bitfield is RAZ when the processor is configured without the Armv8-M Security Extension.

5.27 TRCDEVARCH, Device Architecture Register

The TRCDEVARCH identifies ETM-M85 as an ETMv4.5 component.

Usage constraints

This register is read-only.

Configurations

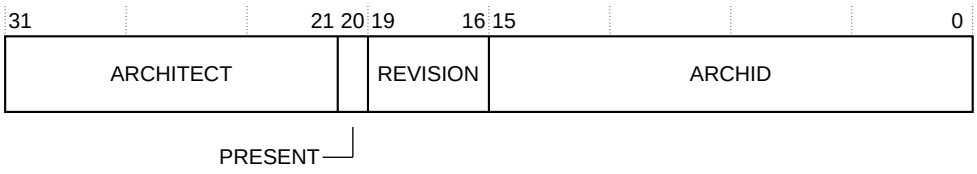
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCDEVARCH bit assignments.

Figure 5-42: TRCDEVARCH bit assignments



The following table shows the TRCDEVARCH bit assignments.

Table 5-46: TRCDEVARCH bit assignments

Bits	Name	Function
[31:21]	ARCHITECT	Defines the architect of the component: **0x23B** Arm.
[20]	PRESENT	Indicates the presence of this register: **0b1** Register is present.
[19:16]	REVISION	Architecture revision: **0x5** Architecture revision 4.5.
[15:0]	ARCHID	Architecture ID: **0x4A13** ETMv4.5 component.

5.28 TRCDEVID, Device ID Register

The TRCDEVID is reserved, **RES0**.

5.29 TRCDEVTYPE, Device Type Register

The TRCDEVTYPE indicates the type of the component.

Usage constraints

This register is read-only.

Configurations

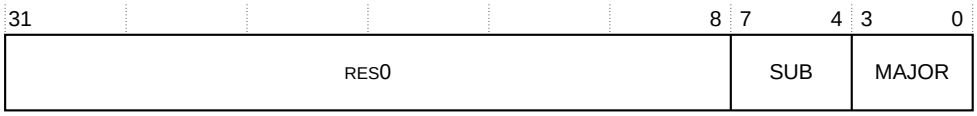
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the TRCDEVTYPE bit assignments.

Figure 5-43: TRCDEVTYPE bit assignments



The following table shows the TRCDEVTYPE bit assignments.

Table 5-47: TRCDEVTYPE bit assignments

Bits	Name	Function
[31:8]	-	**RES0**.
[7:4]	SUB	The subtype of the component: **0b0001** Processor trace.
[3:0]	MAJOR	The main type of the component: **0b0011** Trace source.

5.30 TRCPIDR0-7, Peripheral Identification Registers

The TRCPIDR0-7 provides the standard Peripheral ID required by all CoreSight™ components.

Usage constraints

Only bits[7:0] of each register are used. This means that TRCPIDR0-7 define a single 64-bit *Peripheral ID*, as the following figure shows.

Configurations

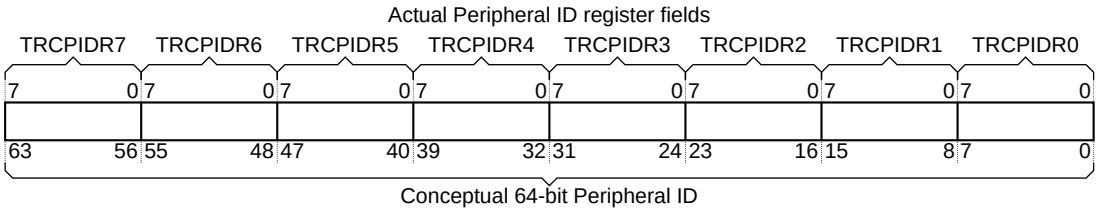
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

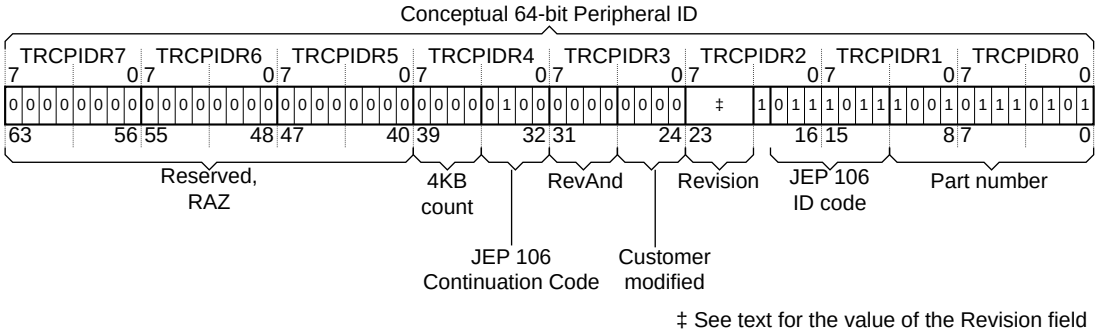
The following figure shows the mapping between TRCPIDR0-7 and the single 64-bit *Peripheral ID* value.

Figure 5-44: Mapping between TRCPIDR0-7 and the Peripheral ID value



The following figure shows the Peripheral ID bit assignments in the single conceptual Peripheral ID register.

Figure 5-45: Peripheral ID fields



The following table shows the values of the fields when reading this set of registers. The *Arm® Embedded Trace Macrocell Architecture Specification ETMv4* gives more information about many of these fields.

The registers are listed in order of register name, from most significant (TRCPIDR7) to least significant (TRCPIDR0). This does not match the order of the register offsets.

Table 5-48: TCRPIDR0-7 bit assignments

Register	Register number	Register offset	Bits	Value	Description
TRCPIDR7	1015	0xFDC	[31:8]	-	**RES0**.
			[7:0]	0x00	**RES0**.
TRCPIDR6	1014	0xFD8	[31:8]	-	**RES0**.
			[7:0]	0x00	**RES0**.
TRCPIDR5	1013	0xFD4	[31:8]	-	**RES0**.
			[7:0]	0x00	**RES0**.
TRCPIDR4	1012	0xFD0	[31:8]	-	**RES0**.
			[7:4]	0x0	n, where 2^n is number of 4KB blocks used.
			[3:0]	0x4	JEP 106 continuation code.
TRCPIDR3	1019	0xFEC	[31:8]	-	**RES0**.
			[7:4]	0x0	RevAnd (at top level). Manufacturer revision number. ECOREVNUM[31:28].
			[3:0]	0x0	Customer Modified. 0x0 indicates from Arm.
TRCPIDR2	1018	0xFE8	[31:8]	-	**RES0**.
			[7:4]	See the Description column for more information.	Revision Number of Peripheral. This value is the same as the Implementation revision field of the TRCIDR1, see 5.16.2 TRCIDR1, ID Register 1 on page 52.
			[3]	0x1	Always 1. Indicates that a JEDEC assigned value is used.
			[2:0]	0x3	JEP 106 identity code [6:4].
TRCPIDR1	1017	0xFE4	[31:8]	-	**RES0**.
			[7:4]	0xB	JEP 106 identity code [3:0].
			[3:0]	0xD	Part Number[11:8].
TRCPIDR0	1016	0xFE0	[31:8]	-	**RES0**.
			[7:0]	0x23	Part Number [7:0].

5.31 TRCCIDR0-3, Component Identification Registers

The TRCCIDR0-3 identifies ETM-M85 as a CoreSight™ component.

Usage constraints

Only bits[7:0] of each register are used. This means that TRCCIDR0-3 define a single 32-bit Component ID, as the following figure shows.

Configurations

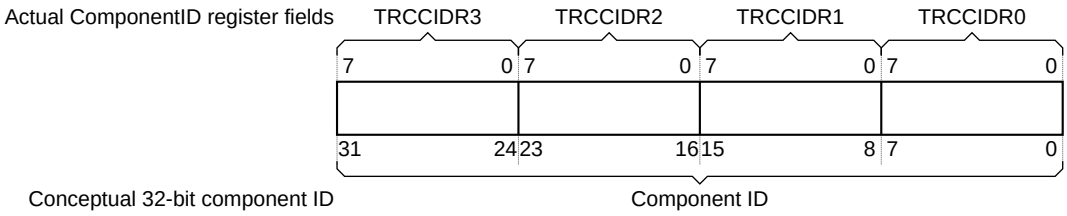
Available in all configurations.

Attributes

See the register summary in Table 5-1: ETM-M85 register summary on page 34.

The following figure shows the mapping between TRCCIDR0-3 and the single 64-bit *Component ID* value.

Figure 5-46: Mapping between TRCCIDR0-3 and the Component ID value



The following table shows the Component ID bit assignments in the single conceptual Component ID register.

The registers are listed in order of register name, from most significant (TRCCIDR3) to least significant (TRCCIDR0). This does not match the order of the register offsets.

Table 5-49: TRCCIDR0-3 bit assignments

Register	Register number	Register offset	Bits	Value	Description
TRCCIDR3	0x3FF	0xFFC	[31:8]	-	**RES0**.
			[7:0]	0xB1	Component identifier, bits[31:24].
TRCCIDR2	0x3FE	0xFF8	[31:8]	-	**RES0**.
			[7:0]	0x05	Component identifier, bits[23:16].
TRCCIDR1	0x3FD	0xFF4	[31:8]	-	**RES0**.
			[7:4]	0x9	Debug component with CoreSight™-compatible registers (component identifier, bits[15:12]).
			[3:0]	0x0	Component identifier, bits[11:8].
TRCCIDR0	0x3FC	0xFF0	[31:8]	-	**RES0**.
			[7:0]	0x0D	Component identifier, bits[7:0].

Appendix A Revisions

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

A.1 Revisions

The following tables show any significant technical changes between released issues of this book.

Table A-1: Issue 0000-02

Change	Location
First release for r0p0	-

Table A-2: Differences between issue 0000-02 and issue 0000-03

Change	Location
First limited access release for r0p0	-
Editorial changes	Throughout document
Progressive terminology commmitment renamed to Inclusive language commitment and editorial content updates made	Document front matter
Information about PMC-100 documentation added	1.1.4 Additional reading on page 10
Incorrect spelling fixed for resource selector 2 in figure Last paragraph updated and expanded to include information about PE comparators	3.3.6 Resource selection on page 29
Incorrect bit assignments diagram fixed	5.16.4 TRCIDR3, ID Register 3 on page 54

Table A-3: Differences between issue 0000-03 and issue 0001-04

Change	Location
Updated processor revision number to r0p1	-
First early access release for r0p1	-
Editorial changes	Throughout document
Added new CTI connections table	3.2 External input and output connections on page 22
Previous topic "Dual issue and fusion of instructions" renamed and updated	3.3.3 Parallel instruction execution on page 28
Last paragraph describing single-shot triggering for DWT PE compare matching removed	3.3.6 Resource selection on page 29
Implementation revision number value updated in REVISION bit field	5.16.2 TRCIDR1, ID Register 1 on page 52

Table A-4: Differences between issue 0001-04 and issue 0002-05

Change	Location
Updated processor revision number to r0p2	-
First release for r0p2	-
Updated the product name to Cortex-M85	Throughout document