Arm® CoreSight™ TPIU-M

Revision: r0p0

Technical Reference Manual

arm

102427_0000_01_en

Arm® CoreSight™ TPIU-M

Technical Reference Manual

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-01	18 May 2021	Non-Confidential	First release for r0p0 EAC

Non-Confidential Proprietary Notice

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

Web Address

developer.arm.com

Progressive terminology commitment

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

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

If you find offensive terms in this document, please contact *terms@arm.com*.

Contents
Arm® CoreSight™ TPIU-M Technical Reference Manual

Introduction

This preface introduces the *Arm® CoreSight™ TPIU-M Technical Reference Manual*.

It contains the following:

About this book

This book describes the interfaces, configuration options, functionality, and programmer's model of the CoreSight™ Trace Port Interface Unit for Cortex®-M processors.

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.

Intended audience

This book is written for the following audiences:

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

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

Using this book

This book is organized into the following chapters:

Chapter 1 About the CoreSight™ TPIU-M
This chapter introduces the CoreSight TPIU-M.

Chapter 2 TPIU-M functional description
This chapter describes the functionality of the CoreSight TPIU-M.

Chapter 3 Programmers model
This chapter describes the programmers model for the CoreSight TPIU-M.

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

Glossary

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

See the *Arm® Glossary* for more information.

Typographic conventions

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

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

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

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

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

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

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

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

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

Timing diagrams

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

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

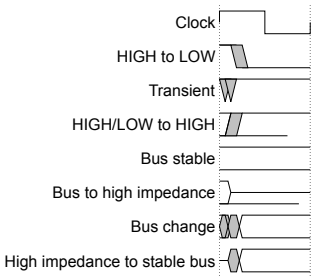

Figure 1 Key to timing diagram conventions

Signals

The signal conventions are:

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

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

Additional reading

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

Arm publications

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

- *Arm® CoreSight™ Architecture Specification v3.0* (IHI 0029E)
- *Arm® CoreSight™ DAP-Lite2 Technical Reference Manual* (100572)
- *AMBA® APB Protocol Specification Version 2.0* (IHI 0024C)
- *AMBA® 4 ATB Protocol Specification ATBv1.0 and ATBv1.1* (IHI 0032B)
- *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces* (IHI 0068C)
- *Arm® v8-M Architecture Reference Manual* (DDI 0553B.o)
- The Technical Reference Manual for your Cortex-M processor

The following confidential books are only available to licensees:

- *Arm® CoreSight™ TPIU-M Configuration and Integration Manual* (102428)
- *Arm® CoreSight™ DAP-Lite2 Configuration and Integration Manual* (100589)
- The Configuration and Integration Manual for your Cortex-M processor

Other publications

- *Verilog-2001 Standard* (IEEE Std 1364-2001)
- Accellera, *IP-XACT version 1685-2009*

Feedback

Feedback on this product

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

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

Feedback on content

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

- The title *Arm CoreSight TPIU-M Technical Reference Manual*.
- The number 102427_0000_01_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

——— **Note** ———

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

———————————

Chapter 1
About the CoreSight™ TPIU-M

This chapter introduces the CoreSight TPIU-M.

It contains the following sections:

1.1 About this product

CoreSight TPIU-M is a *Trace Port Interface Unit* (TPIU) that is designed for use in single-processor systems based on Arm Cortex-M processors. Using TPIU-M, you can export instrumentation trace data and processor execution trace data off-chip to a *Trace Port Analyzer* (TPA) for use in a software debugging environment, for example Arm Development Studio or Keil® *Microcontroller Development Kit* (MDK).

You can build a full CoreSight debug and trace system for a single Arm Cortex-M processor using:

- DAP-Lite2 to provide debug connectivity and control
- TPIU-M to export trace data off chip

Other CoreSight products are available to support designs with multiple processors. Contact Arm for further details.

1.2 Features

TPIU-M supports the following features:

- 8-bit ATB interface that supports *Instrumentation Trace Macrocell* (ITM) trace from a Cortex-M processor
- Optional second 8-bit ATB interface that supports *Embedded Trace Macrocell* (ETM) trace from a Cortex-M processor
- Configurable width parallel trace port interface for connection to an off-chip *Trace Port Analyzer* (TPA)
- Serial Wire Output support for connection to low-cost TPAs using a single pin
- Programmable clock prescaler for both parallel and serial trace output modes
- Two asynchronous clock domains: APB/ATB and trace port
- Q-Channel *Low-Power Interfaces* (LPIs) to support low-power implementation
- Low gate count

Note

TPIU-M supports only one trace source per ATB interface. You must connect the TPIU-M ATB interfaces directly to the Cortex-M ITM and ETM. TPIU-M is optimised for direct connection only. You must not instantiate bridges, replicators, funnels, or other components on the ATB paths between the trace sources and the TPIU-M.

1.3 Supported standards

CoreSight TPIU-M is compliant with the following standards:

- *Arm® CoreSight™ Architecture Specification v3.0*
- *AMBA® APB Protocol Specification Version 2.0*
- *AMBA® 4 ATB Protocol Specification ATBv1.0 and ATBv1.1*
- *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces*
- *Verilog-2001 Standard*
- Accellera, *IP-XACT version 1685-2009*

1.4 Documentation

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

Technical Reference Manual

The TRM describes the functionality and the effects of functional options on the behavior of the TPIU-M. It is required at all stages of the design flow. The choices that you make in the design flow can mean that some behaviors that are described in the TRM are not relevant. If you are programming a device that is based on TPIU-M components, then contact:

- The implementer to determine:
  - The build configuration of the implementation
  - The integration, if any, that was performed before implementing the TPIU-M
- The integrator to determine the pin configuration of your device.

Configuration and Integration Manual

The CIM describes:

- How to configure the TPIU-M
- How to integrate the TPIU-M into your Cortex-M processor-based system
- How to implement the TPIU-M components to produce a hard macrocell of the design. This description includes custom cell replacement, a description of the power domains, and a description of the design synthesis.

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

1.5 Design process

The TPIU-M is delivered as synthesizable Verilog RTL.

Before the TPIU-M can be used in a product, it must go through the following processes:

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

Configuration
Defining the memory map of the system and the functional configuration of the TPIU-M components.

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

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

Implementation
Using the Verilog RTL in an implementation flow to produce a hard macrocell.

The operation of the final device depends on:

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

Software configuration
The programmer configures the CoreSight debug and trace subsystem by programming specific values into registers that affect the behavior of the TPIU-M components.

1.6 TPIU-M component

The following table shows the component and its version.

Table 1-1 TPIU-M component list

Name	Description	Version	Revision	IP-XACT version
`tpium`	Trace Port Interface Unit for Cortex-M	r0p0	0	r0p0_0

Note

The Revision column shows the value of the PIDR2.REVISION field.

1.7 Product revisions

This section describes the differences in functionality between product revisions of the CoreSight TPIU-M.

r0p0

First release of CoreSight TPIU-M.

Chapter 2
TPIU-M functional description

This chapter describes the functionality of the CoreSight TPIU-M.

It contains the following sections:

2.1 Functional interfaces

This section describes the functional interfaces of the TPIU-M.

The functional interfaces are:

ATB slave interfaces
Receive trace data

APB slave interface
Accesses the TPIU-M registers

Trace out port
Connects to the external trace port pins

Serial Wire Output
Connects to the external SWO pin

Triggers from Cortex-M processor
Permit interaction with the processor ITM and DWT counters and ETM events

The following figure shows the external connections of the TPIU-M.

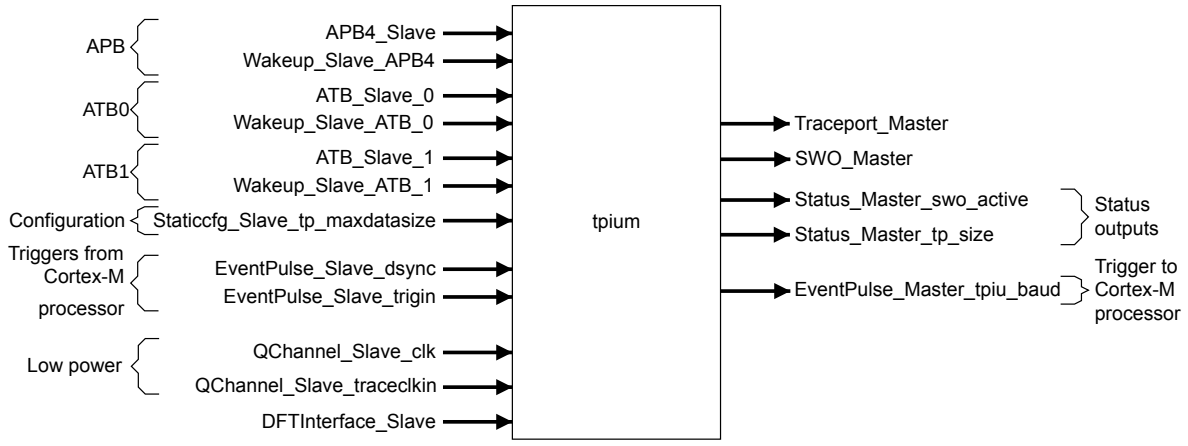

Figure 2-1 TPIU-M block diagram

2.2 Clocks and resets

The clock and reset signals of the TPIU-M are **clk**, **traceclkin**, **reset_n**, and **treset_n**.

The TPIU-M includes an asynchronous bridge between the **traceclkin** clock domain and the rest of the design.

2.3 Trace output modes

TPIU-M supports three trace output modes:

- Parallel trace mode
- *Serial Wire Output* (SWO) using UART (NRZ) encoding
- SWO using Manchester encoding

To select the trace output modes, set TPIU_SPPR.TXMODE.

The Parallel trace is synchronously clocked. TPIU-M supports 16, 12, 8, 4, 2, and 1-bit data widths, set by the system integrator. TPIU_SSPSR register describes the supported widths.

To set the active parallel trace width, program the TPIU_CSPSR register. This enables debug tools to select a value that is supported by their *Trace Port Analyzer* (TPA).

For all three trace output modes, the data rate is a function of **traceclkin** and the clock prescaler `TPIU_ACPR`.

2.4 Effect of the trace clock prescaler

The TPIU-M includes a 13-bit prescaler register TPIU_ACPR that you can program to slow the rate of generated trace compared to the reference clock **traceclkin**.

The prescaler affects both Serial Wire Output and Parallel trace modes. It determines the symbol rate of the TPIU-M output interfaces as follows:

symbol rate = **traceclkin** * 1/(`TPIU_ACPR.SWOSCALER` + 1)

The following figure shows the prescaler timing.

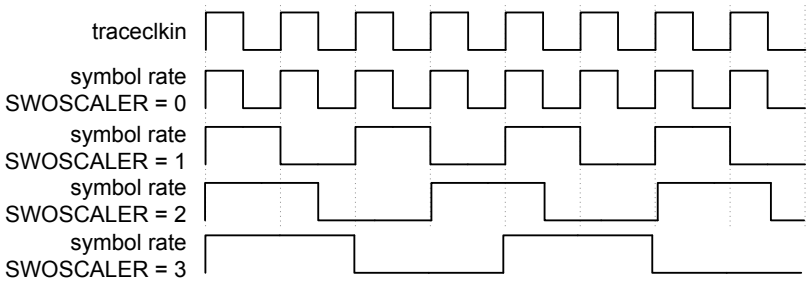

Figure 2-2 TPIU-M prescaler timing

2.5 Parallel trace output mode

When the TPIU-M is programmed to generate parallel trace, when `TPIU_SPPR.TXMODE` = `0b00`:

- **tracedata** can change at a maximum rate of the symbol rate
- **traceclk** changes at the symbol rate

The frequency of **traceclk** is therefore half of the symbol rate.

Trace data is sampled on both edges of **traceclk**.

The gross bit rate of the parallel interface is N × symbol rate, where N is the number of parallel bits.

The following figure shows the parallel trace timing.

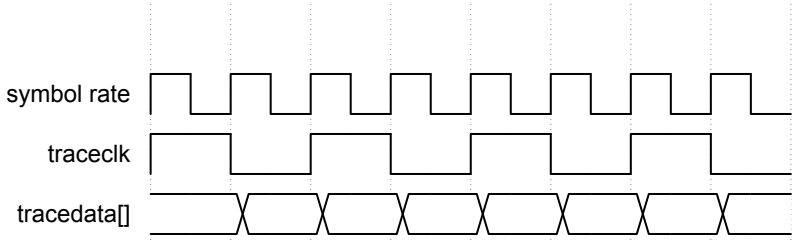

Figure 2-3 TPIU-M parallel trace timing

2.6 SWO modes

In SWO modes the **traceclk** signal remains static.

UART (NRZ) mode

When the TPIU-M is programmed to generate UART *Non Return to Zero* (NRZ) format SWO, when `TPIU_SPPR.TXMODE = 0b10`, swo can change at the symbol rate.

The gross bit rate of the interface is equal to 1 * symbol rate.

UART (NRZ) encoding has an overhead of two symbols that represent the start bit and stop bit for every 8 data bits.

The following figure shows the SWO UART timing.

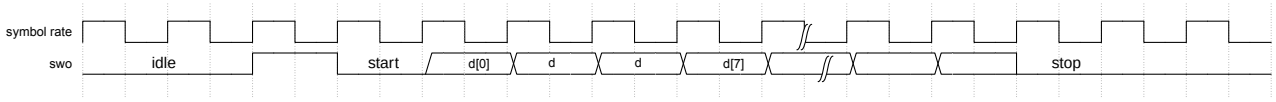

Figure 2-4 TPIU-M SWO UART timing

Manchester mode

When the TPIU-M is programmed to generate Manchester format SWO, when `TPIU_SPPR.TXMODE` = 0b01, swo may change at the symbol rate.

The gross bit rate of the interface is equal to symbol rate / 2, because each data bit is represented by two symbols.

Manchester encoding has an overhead of four symbols that represent the start and stop bit for each block of 8-64 data bits. The data is sent LSB first.

The following figure shows the SWO Manchester timing.

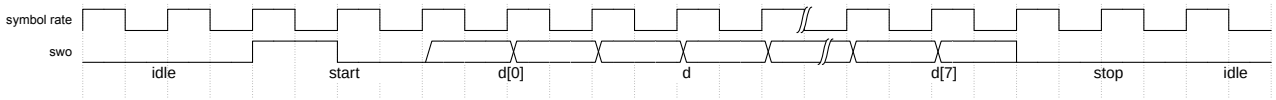

Figure 2-5 TPIU-M SWO Manchester timing

2.7 Output interfaces

This section contains the following subsections:

2.7.1 Parallel trace out port

This section describes the parallel trace out port signals.

To select parallel trace mode you must program TPIU_SPPR. TPIU_SSPSR indicates the supported parallel trace port widths. To select the active parallel trace port width, program TPIU_CSPSR. You can vary the data rate of the interface by programming TPIU_ACPR.

The following table summarizes the trace out port signals.

Table 2-1 Trace out port signals

Signal	Type	Description
traceclk	Output	Output clock, that the TPA uses to sample the other pins of the trace out port. This signal is **traceclkin** divided by 2*(TPIU_ACPR.SWOSCALER +1) when parallel trace is enabled (TPIU_SPPR=0x0) and otherwise static, and data is valid on both edges of this clock.
tracedata[TRACEPORT_DATA_WIDTH-1:0]	Output	Output data. A system might not connect all the bits of this signal to the trace port pins. The connection depends on the number of pins available and the bandwidth that is required to output trace.

2.7.2 Serial Wire Output

This section describes the Serial Wire Output signals.

To select Serial Wire Output mode you must program TPIU_SPPR. You can vary the data rate of the interface by pogramming TPIU_ACPR.

The following table summarizes the Serial Wire output signals.

Table 2-2 Trace out port signals

Signal	Type	Description
swo	Output	Serial Wire Output data. Connect this signal to a pin on your device, for connection to a Trace Port Analyser.

2.8 traceclk alignment

The TPIU-M does not offset the edges of **traceclk** from the edges of the trace data signals **tracedata**.

Arm recommends that, to support the widest range of targets at the maximum speed, TPAs support systems with a variety of alignments of **traceclk** relative to the data signals, including systems where edges of **traceclk** occur at the same time as transitions of the data signals.

2.9 Trace port triggers

TPIU-M supports Continuous mode and Bypass mode only.

See *TPIU Formatter and Flush Control Register, TPIU_FFCR*.

Triggers received by the TPIU-M on the **trigin** input cause trigger packets to be inserted into the formatted trace data in Continuous mode. Triggers are ignored in Bypass mode.

2.10 Programming the TPIU-M for trace capture

Arm recommends that debug tools perform a flush by writing TPIU_FFCR.FOnMan = 1 whenever the TPIU-M is reprogrammed. This causes a Full Sync Packet to be generated in the formatted trace stream, which enables the connected *Trace Port Analyzer* (TPA) to detect the start the frame and decode the subsequent data.

2.11 Example configuration scenarios

You can choose from the following modes, controlled with TPIU_SPPR:

- Asynchronous trace via **swo**, Manchester encoded: rate is set with TPIU_ACPR
- Asynchronous trace via **swo**, NRZ encoded: rate is set with TPIU_ACPR
- Parallel Trace via **tracedata**: width is set with TPIU_CSPSR, rate is set with TPIU_ACPR

Chapter 3
Programmers model

This chapter describes the programmers model for the CoreSight TPIU-M.

It contains the following sections:

3.1 TPIU-M register summary

The register summary lists all the TPIU-M registers and their key characteristics.

Table 3-1 tpium register summary

Offset	Name	Type	Reset	Width	Description
0x000	TPIU_SSPSR	RO	0x0000----	32	*3.2 TPIU Supported Parallel Port Sizes Register, TPIU_SSPSR* on page 3-35
0x004	TPIU_CSPSR	RW	0x00000001	32	*3.3 TPIU Current Parallel Port Size Register, TPIU_CSPSR* on page 3-36
0x010	TPIU_ACPR	RW	0x00000000	32	*3.4 TPIU Asynchronous Clock Prescaler Register, TPIU_ACPR* on page 3-37
0x0F0	TPIU_SPPR	RW	0x00000001	32	*3.5 TPIU Selected Pin Protocol Register, TPIU_SPPR* on page 3-39
0x300	TPIU_FFSR	RO	0x00000008	32	*3.6 TPIU Formatter and Flush Status Register, TPIU_FFSR* on page 3-40
0x304	TPIU_FFCR	RW	0x00000100	32	*3.7 TPIU Formatter and Flush Control Register, TPIU_FFCR* on page 3-41
0x308	TPIU_PSCR	RW	0x0000000A	32	*3.8 TPIU Periodic Synchronization Control Register, TPIU_PSCR* on page 3-43
0xEE4	TPIU_ITDSYNC	RO	0x00000000	32	*3.9 TPIU Integration Test DSYNC Register, TPIU_ITDSYNC* on page 3-45
0xEE8	TPIU_ITTRIGGER	RO	0x00000000	32	*3.10 TPIU Integration Test Trigger Register, TPIU_ITTRIGGER* on page 3-46
0xEEC	TPIU_ITFTTD0	RO	0x00000000	32	*3.11 TPIU Integration Test FIFO Test Data Register 0, TPIU_ITFTTD0* on page 3-47
0xEF0	TPIU_ITATBCTR2	WO	0x00000000	32	*3.12 TPIU Integration Test ATB Control Register 2, TPIU_ITATBCTR2* on page 3-49
0xEF4	TPIU_ITATBCTR1	RO	0x00000000	32	*3.13 Integration Test ATB Control Register 1, TPIU_ITATBCTR1* on page 3-50
0xEF8	TPIU_ITATBCTR0	RO	0x00000000	32	*3.14 Integration Test ATB Control Register 0, TPIU_ITATBCTR0* on page 3-51
0xEFC	TPIU_ITFTTD1	RW	0x00000000	32	*3.15 TPIU Integration Test FIFO Test Data Register 1, TPIU_ITFTTD1* on page 3-52
0xF00	TPIU_ITCTRL	RW	0x00000000	32	*3.16 TPIU Integration Mode Control Register, TPIU_ITCTRL* on page 3-54
0xFA0	TPIU_CLAIMSET	RW	0x0000000F	32	*3.17 TPIU Claim Tag Set Register, TPIU_CLAIMSET* on page 3-56
0xFA4	TPIU_CLAIMCLR	RW	0x00000000	32	*3.18 TPIU Claim Tag Clear Register, TPIU_CLAIMCLR* on page 3-57
0xFA8	TPIU_DEVAFF0	RO	0x00000000	32	*3.19 TPIU Device Affinity register 0, TPIU_DEVAFF0* on page 3-58
0xFAC	TPIU_DEVAFF1	RO	0x00000000	32	*3.20 TPIU Device Affinity register 1, TPIU_DEVAFF1* on page 3-59
0xFBC	TPIU_DEVARCH	RO	0x00000000	32	*3.21 TPIU Device Architecture Register, TPIU_DEVARCH* on page 3-60
0xFC0	TPIU_DEVID2	RO	0x00000000	32	*3.22 TPIU Device Configuration Register 2, TPIU_DEVID2* on page 3-61

Table 3-1 tpium register summary (continued)

Offset	Name	Type	Reset	Width	Description
0xFC4	TPIU_DEVID1	RO	0x00000000	32	*3.23 TPIU Device Configuration Register 1, TPIU_DEVID1* on page 3-62
0xFC8	TPIU_DEVID	RO	0x00020C2-	32	*3.24 TPIU Device Identifier Register, TPIU_DEVID* on page 3-63
0xFCC	TPIU_DEVTYPE	RO	0x00000011	32	*3.25 TPIU Device Type Register, TPIU_DEVTYPE* on page 3-65
0xFD0	TPIU_PIDR4	RO	0x00000004	32	*3.26 TPIU Peripheral Identification Register 4, TPIU_PIDR4* on page 3-66
0xFD4	TPIU_PIDR5	RO	0x00000000	32	*3.27 TPIU Peripheral Identification Register 5, TPIU_PIDR5* on page 3-67
0xFD8	TPIU_PIDR6	RO	0x00000000	32	*3.28 TPIU Peripheral Identification Register 6, TPIU_PIDR6* on page 3-68
0xFDC	TPIU_PIDR7	RO	0x00000000	32	*3.29 TPIU Peripheral Identification Register 7, TPIU_PIDR7* on page 3-69
0xFE0	TPIU_PIDR0	RO	0x000000F1	32	*3.30 TPIU Peripheral Identification Register 0, TPIU_PIDR0* on page 3-70
0xFE4	TPIU_PIDR1	RO	0x000000B9	32	*3.31 TPIU Peripheral Identification Register 1, TPIU_PIDR1* on page 3-71
0xFE8	TPIU_PIDR2	RO	0x0000000B	32	*3.32 TPIU Peripheral Identification Register 2, TPIU_PIDR2* on page 3-72
0xFEC	TPIU_PIDR3	RO	0x00000000	32	*3.33 TPIU Peripheral Identification Register 3, TPIU_PIDR3* on page 3-73
0xFF0	TPIU_CIDR0	RO	0x0000000D	32	*3.34 TPIU Component Identification Register 0, TPIU_CIDR0* on page 3-74
0xFF4	TPIU_CIDR1	RO	0x00000090	32	*3.35 TPIU Component Identification Register 1, TPIU_CIDR1* on page 3-75
0xFF8	TPIU_CIDR2	RO	0x00000005	32	*3.36 TPIU Component Identification Register 2, TPIU_CIDR2* on page 3-76
0xFFC	TPIU_CIDR3	RO	0x000000B1	32	*3.37 TPIU Component Identification Register 3, TPIU_CIDR3* on page 3-77

3.2 TPIU Supported Parallel Port Sizes Register, TPIU_SSPSR

Indicates the supported parallel trace port sizes.

The possible values of each bit are:

0
Parallel trace port width (m+1) not supported.

1
Parallel trace port width (m+1) supported.

Attributes

Offset
`0x000`

Type
Read-only

Reset
`0x0000----`

Width
32

Bit descriptions

The following image shows the register bit assignments.

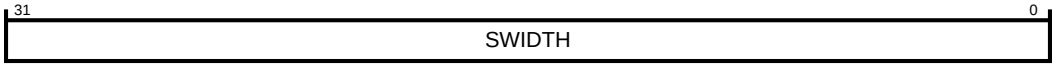

Figure 3-1 TPIU_SSPSR

The following table shows the register bit assignments.

Table 3-2 TPIU_SSPSR attributes

Bits	Reset value	Name	Type	Function
[31:0]	IMPLEMENTATION DEFINED	SWIDTH	Read-only	Supported width. SWIDTH[m] indicates whether a parallel trace port width of (m+1) is supported. The supported port sizes depend on the configuration parameter TRACEPORT_DATA_WIDTH and the configuration tie-off **tp_maxdatasize**. **0x0001** 1 bit Parallel Trace is supported. **0x0003** 1 and 2 bit Parallel Trace is supported. **0x000B** 1, 2 and 4 bit Parallel Trace is supported. **0x008B** 1, 2, 4 and 8 bit Parallel Trace is supported. **0x088B** 1, 2, 4, 8 and 12 bit Parallel Trace is supported. **0x888B** 1, 2, 4, 8, 12 and 16 bit Parallel Trace is supported.

3.3 TPIU Current Parallel Port Size Register, TPIU_CSPSR

Controls the width of the parallel trace port.

The possible values of each bit are:

0
Width (m+1) is not the current parallel trace port width.

1
Width (m+1) is the current parallel trace port width.

A debugger must set only one bit to 1, and all others must be zero.

The effect of writing a value with more than one bit set to 1 is architecurally UNPREDICTABLE.

The effect of a write to an unsupported bit is UNPREDICTABLE.

This register resets to the value for the smallest supported parallel trace port size.

Attributes

Offset
`0x004`

Type
Read-write

Reset
`0x00000001`

Width
32

Bit descriptions

The following image shows the register bit assignments.

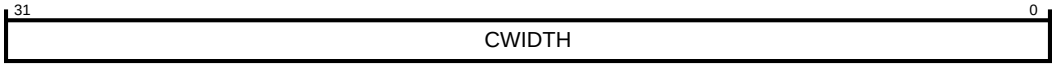

Figure 3-2 TPIU_CSPSR

The following table shows the register bit assignments.

Table 3-3 TPIU_CSPSR attributes

Bits	Reset value	Name	Type	Function
[31:0]	0x1	CWIDTH	Read-write	Current width. CWIDTH[m] represents a parallel trace port width of (m+1).

3.4 TPIU Asynchronous Clock Prescaler Register, TPIU_ACPR

Defines a prescaler value for the baud rate of the Serial Wire Output (SWO).

Writing to the register automatically updates the prescale counter, immediately affecting the baud rate of the serial data output.

If a debugger changes the register value while the TPIU is transmitting data, the effect on the output stream is UNPREDICTABLE and the required recovery process is IMPLEMENTATION DEFINED.

SWO or Parallel trace port output clock = Asynchronous_Reference_Clock/($n + 1)

When TPIU_SPPR.TXMODE=0b00, the parallel trace clock traceclk is traceclkin / 2*(TPIU_ACPR.SWOSCALER + 1).

When TPIU_SPPR.TXMODE=0b01 or TPIU_SPPR.TXMODE=0b10 the Serial Wire Output trace is generated at a rate of traceclkin / (TPIU_ACPR.SWOSCALER + 1), and the parallel trace clock traceclk remains static.

Attributes

Offset

`0x010`

Type

Read-write

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

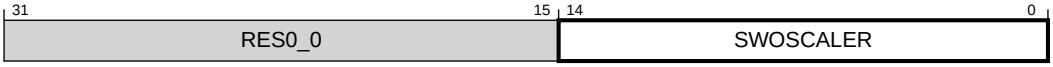

Figure 3-3 TPIU_ACPR

The following table shows the register bit assignments.

Table 3-4 TPIU_ACPR attributes

Bits	Reset value	Name	Type	Function
[31:15]	0x0	RES0_0	Read-write	Reserved bit or field with SBZP behavior
[14:0]	0x0	SWOSCALER	Read-write	SWO and Parallel trace port baud rate prescaler. Sets the ratio between an IMPLEMENTATION DEFINED reference clock and the TPIU output clock rates. The prescaler always sets the ratio for the SWO output clock. When TPIU_DEVID.CPPT is one, the prescaler also sets the ratio for the Parallel trace port clock. The supported scaler value range is IMPLEMENTATION DEFINED, to a maximum scaler value of `0xFFFF`. Unused bits of this field are RAZ/WI. TPIU-M implements bits [12:0] only, supporting a maximum scaler value of `0x1FFF`.

3.5 TPIU Selected Pin Protocol Register, TPIU_SPPR

Selects the protocol used for trace output.

The effect of selecting a reserved value, or a mode that the implementation does not support, is UNPREDICTABLE.

If a debugger changes the register value while the TPIU is transmitting data, the effect on the output stream is UNPREDICTABLE and the required recovery process is IMPLEMENTATION DEFINED.

Attributes

Offset

`0x0F0`

Type

Read-write

Reset

`0x00000001`

Width

32

Bit descriptions

The following image shows the register bit assignments.

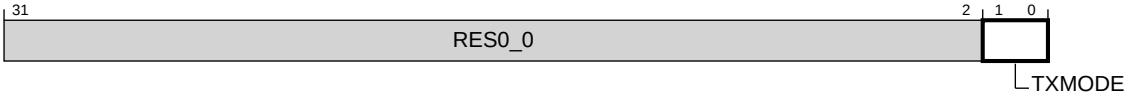

Figure 3-4 TPIU_SPPR

The following table shows the register bit assignments.

Table 3-5 TPIU_SPPR attributes

Bits	Reset value	Name	Type	Function
[31:2]	0x0	RES0_0	Read-write	Reserved bit or field with SBZP behavior
[1:0]	0b01	TXMODE	Read-write	Transmit mode. Specifies the protocol for trace output from the TPIU. **0b00** Parallel trace port mode. This value is reserved if TPIU_DEVID.PTINVALID == 1. **0b01** Asynchronous SWO, using Manchester encoding. This value is reserved if TPIU_DEVID.MANCVALID == 0. **0b10** Asynchronous SWO, using NRZ encoding. This value is reserved if TPIU_DEVID.NRZVALID == 0. **0b11** RESERVED.

3.6 TPIU Formatter and Flush Status Register, TPIU_FFSR

Shows the status and capabilities of the TPIU formatter.

Attributes

Offset

`0x300`

Type

Read-only

Reset

`0x00000008`

Width

32

Bit descriptions

The following image shows the register bit assignments.

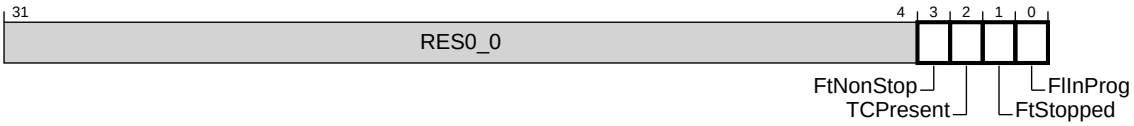

Figure 3-5 TPIU_FFSR

The following table shows the register bit assignments.

Table 3-6 TPIU_FFSR attributes

Bits	Reset value	Name	Type	Function
[31:4]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[3]	1	FtNonStop	Read-only	Non-stop formatter. The value cannot be changed. The fixed value of 1 indicates that the formatter cannot be stopped.
[2]	0	TCPresent	Read-only	TRACECTL present. The value cannot be changed. The fixed value of 0 indicates that the TRACECTL pin is not present.
[1]	0	FtStopped	Read-only	Formatter stopped. The value cannot be changed. The fixed value of 0 indicates that the formatter has not stopped.
[0]	0	FlInProg	Read-only	Flush in progress. Set to 1 when a flush is initiated and clears to zero when all data received before the flush is acknowledged has been output on the trace port. That is, the trace has been received at the sink, formatted, and output on the trace port. **0** No ongoing flush. **1** Flush in progress.

3.7 TPIU Formatter and Flush Control Register, TPIU_FFCR

Controls the TPIU formatter.

Attributes

Offset

`0x304`

Type

Read-write

Reset

`0x00000100`

Width

32

Bit descriptions

The following image shows the register bit assignments.

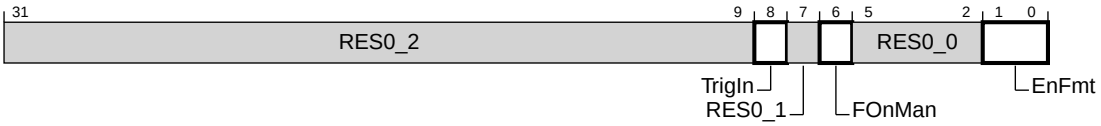

Figure 3-6 TPIU_FFCR

The following table shows the register bit assignments.

Table 3-7 TPIU_FFCR attributes

Bits	Reset value	Name	Type	Function
[31:9]	0x0	RES0_2	Read-write	Reserved bit or field with SBZP behavior
[8]	1	TrigIn	Read-only	Trigger input asserted. Indicate a trigger on the trace port when an IMPLEMENTATION DEFINED TRIGIN signal is asserted. **0** Disable trigger indication when trigin is asserted. **1** Enable trigger indication when trigin is asserted.
[7]	0	RES0_1	Read-write	Reserved bit or field with SBZP behavior
[6]	0	FOnMan	Read-write	Flush On Manual. Setting this bit to 1 generates a flush. The TPIU clears the bit to 0 when the flush completes.

Table 3-7 TPIU_FFCR attributes (continued)

Bits	Reset value	Name	Type	Function
[5:2]	0b0000	RES0_0	Read-write	Reserved bit or field with SBZP behavior
[1:0]	0b00	EnFmt	Read-write	Formatter control. Selects the output formatting mode. This field must be set to 0b10 when the parallel trace port is selected, or when using multiple trace sources. Changing the value of this field when TPIU_FFSR.FtStopped is 0 is UNPREDICTABLE. Arm recommends that you change this field only when the ATB interfaces are idle. **0b00** Bypass. Disable formatting. Only supported when SWO mode is selected. Only a single trace source is supported in bypass mode: If only a single trace source is connected to this TPIU, it is selected. If multiple sources (including the ITM) are implemented and connected to this TPIU, then all other trace sources, except for the ITM, must be disabled. Otherwise, the trace output is UNPREDICTABLE. All other trace sources are discarded. **0b01** Reserved. **0b10** Continuous. Enable formatting and embed triggers and null cycles in the formatted output. **0b11** Reserved.

3.8 TPIU Periodic Synchronization Control Register, TPIU_PSCR

Defines the reload value for the Periodic Synchronization Counter register.

The Periodic Synchronization Counter decrements for each byte that is output by the TPIU. If the formatter is implemented and enabled, the TPIU forces completion of the current frame when the counter reaches zero. The TPIU forces all trace sources to generate synchronization packets with the implemented **syncreqx_s** signals when the counter reaches zero. Bytes generated by the TPIU as part of a Halfword synchronization packet or a Full frame synchronization packet are not counted.

Attributes

Offset

`0x308`

Type

Read-write

Reset

`0x0000000A`

Width

32

Bit descriptions

The following image shows the register bit assignments.

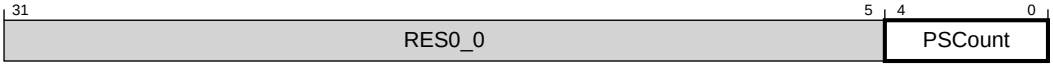

Figure 3-7 TPIU_PSCR

The following table shows the register bit assignments.

Table 3-8 TPIU_PSCR attributes

Bits	Reset value	Name	Type	Function
[31:5]	0x0	RES0_0	Read-write	Reserved bit or field with SBZP behavior
[4:0]	0b01010	PSCount	Read-write	Periodic Synchronization Count. Determines the reload value of the Periodic Synchronization Counter. The reload value takes effect the next time the counter reaches zero. Reads from this register return the reload value programmed into this register.The possible values of this field are: **0b00000** Synchronization disabled. **0b00111** 128 bytes. **0b01000** 256 bytes. **...** **0b11111** 2^31 bytes. All other values are reserved. The Periodic Synchronization Counter might have a maximum value smaller than 2^31. In this case, if the programmed reload value is greater than the maximum value, then the Periodic Synchronization Counter is reloaded with its maximum value and the TPIU will generate synchronization requests at this interval. TPIU-M supports a maximum value of `0b100000`.

3.9 TPIU Integration Test DSYNC Register, TPIU_ITDSYNC

This register indicates the integration status of the dsync input in integration mode.

Reads are allowed even in functional mode, but the register itself is disabled and does not get updated even if the inputs change.

Attributes

Offset

`0xEE4`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

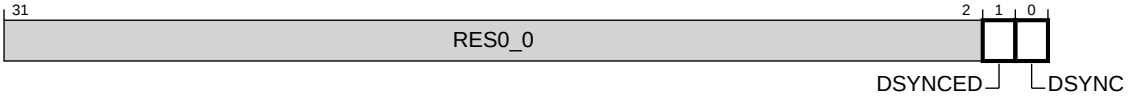

Figure 3-8 TPIU_ITDSYNC

The following table shows the register bit assignments.

Table 3-9 TPIU_ITDSYNC attributes

Bits	Reset value	Name	Type	Function
[31:2]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[1]	0	DSYNCED	Read-only	This bit indicates that dsync was observed HIGH since integration mode was enabled. Reading this register clears the bit if dsync is LOW.
[0]	0	DSYNC	Read-only	This bit returns the value of the dsync input.

3.10 TPIU Integration Test Trigger Register, TPIU_ITTRIGGER

This register indicates the integration status of the trigger input in integration mode.

Reads are allowed even in functional mode, but the register itself is disabled and does not get updated even if the inputs change.

Attributes

Offset

`0xEE8`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

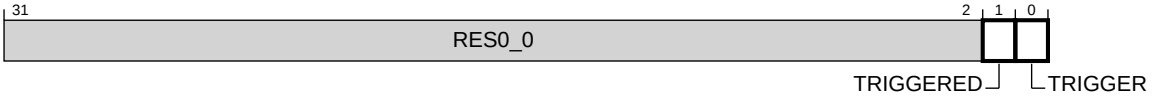

Figure 3-9 TPIU_ITTRIGGER

The following table shows the register bit assignments.

Table 3-10 TPIU_ITTRIGGER attributes

Bits	Reset value	Name	Type	Function
[31:2]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[1]	0	TRIGGERED	Read-only	This bit indicates that trigin was observed HIGH since integration mode was enabled. Reading this register clears the bit if trigin is LOW.
[0]	0	TRIGGER	Read-only	This bit returns the value of the trigin input.

3.11 TPIU Integration Test FIFO Test Data Register 0, TPIU_ITFTTD0

This register indicates the integration status of the ATB data interfaces.

To read this register the integration data test mode must be enabled with TPIU_ITCTRL.Mode=`0b10`.

Reads are allowed otherwise but the register itself is disabled and doesn't get updated even if the inputs change.

Attributes

Offset

`0xEEC`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

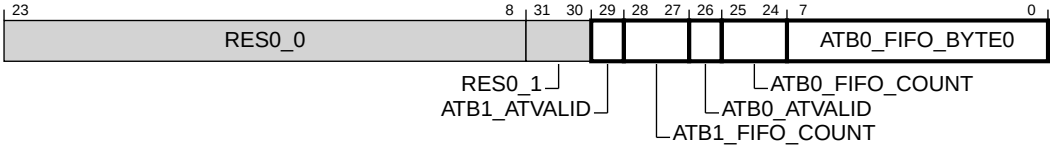

Figure 3-10 TPIU_ITFTTD0

The following table shows the register bit assignments.

Table 3-11 TPIU_ITFTTD0 attributes

Bits	Reset value	Name	Type	Function
[23:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[31:30]	0b00	RES0_1	Read-only	Reserved bit or field with SBZP behavior
[29]	0	ATB1_ATVALID	Read-only	Returns the value of the ATB1 Interface signal **atvalid1_s**. This field is RAZ if the TPIU-M does not include the ATB1 interface.
[28:27]	0b00	ATB1_FIFO_COUNT	Read-only	Number of bytes of ATB1 Interface trace data since last read of this register. This field is RAZ if the TPIU-M does not include the ATB1 interface. **0b00** 0 Bytes received and **atready1_s**=1 **0b01** 1 Byte received and **atready1_s**=0 **0b10** Reserved. **0b11** Reserved.
[26]	0	ATB0_ATVALID	Read-only	Returns the value of the ATB0 Interface signal **atvalid0_s**.

Table 3-11 TPIU_ITFTTD0 attributes (continued)

Bits	Reset value	Name	Type	Function
[25:24]	0b00	ATB0_FIFO_COUNT	Read-only	Number of bytes of ATB0 Interface trace data since last read of this register. **0b00** 0 Bytes received and **atready0_s**=1 **0b01** 1 Byte received and **atready0_s**=0 **0b10** Reserved. **0b11** Reserved.
[7:0]	0x0	ATB0_FIFO_BYTE0	Read-only	ATB0 Interface trace data FIFO byte 0. The TPIU-M discards this data when the register is read and as a consequence ATB0 FIFO byte count clears to `0x0`.

3.12 TPIU Integration Test ATB Control Register 2, TPIU_ITATBCTR2

This register enables control of the ATB control signal outputs in integration mode.

- **atready0_s, atready1_s**
- **afvalid0_s, afvalid1_s**
- **syncreq0_s, syncreq1_s**

Writes to this register are allowed in integration mode as well as functional mode. However, the programmed value is driven to the outputs only in integration mode.Note: ATB1 signals are only driven when TPIU-M includes the ATB1 interface.

Attributes

Offset

`0xEF0`

Type

Write-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

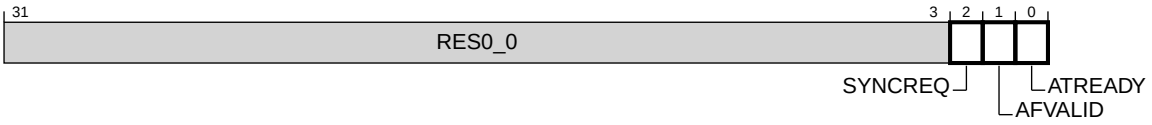

Figure 3-11 TPIU_ITATBCTR2

The following table shows the register bit assignments.

Table 3-12 TPIU_ITATBCTR2 attributes

Bits	Reset value	Name	Type	Function
[31:3]	0x0	RES0_0	Write-only	Reserved bit or field with SBZP behavior
[2]	0	SYNCREQ	Write-only	Sets the value of **syncreq0_s** and **syncreq1_s** in integration mode.
[1]	0	AFVALID	Write-only	Sets the value of **afvalid0_s** and **afvalid1_s** in integration mode.
[0]	0	ATREADY	Write-only	Sets the value of **atready0_s** and **atready1_s** in integration mode.

3.13 Integration Test ATB Control Register 1, TPIU_ITATBCTR1

This register indicates the value of the **atid0_s** and **atid1_s** inputs in integration mode.

Reads are allowed even in functional mode, but the register is disabled and does not get updated even if the inputs change. The reset value depends on external source driving these inputs.

Attributes

Offset

`0xEF4`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

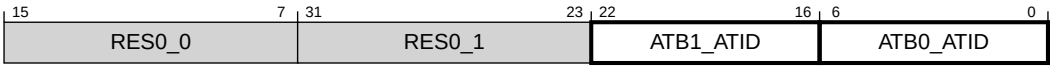

Figure 3-12 TPIU_ITATBCTR1

The following table shows the register bit assignments.

Table 3-13 TPIU_ITATBCTR1 attributes

Bits	Reset value	Name	Type	Function
[15:7]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[31:23]	0x0	RES0_1	Read-only	Reserved bit or field with SBZP behavior
[22:16]	0x0	ATB1_ATID	Read-only	Reads the value of **atid1_s[6:0]** in integration mode.This field is RAZ if the TPIU-M does not include the ATB1 interface.
[6:0]	0x0	ATB0_ATID	Read-only	Reads the value of **atid0_s[6:0]** in integration mode.

3.14 Integration Test ATB Control Register 0, TPIU_ITATBCTR0

This register indicates the values of **atvalid_s**, **afready_s**, and **atwakeup_s** inputs in integration mode.

Reads are allowed even in functional mode, but the register is disabled and does not get updated even if the inputs change. The reset value depends on an external source driving these inputs.

Attributes

Offset

`0xEF8`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

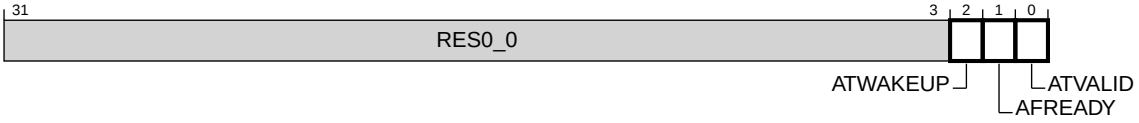

Figure 3-13 TPIU_ITATBCTR0

The following table shows the register bit assignments.

Table 3-14 TPIU_ITATBCTR0 attributes

Bits	Reset value	Name	Type	Function
[31:3]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[2]	0	ATWAKEUP	Read-only	Reads the value of **atwakeup0_s** logically ORed with **atwakeup1_s** in integration mode.
[1]	0	AFREADY	Read-only	Reads the value of **afready0_s** logically ORed with **afready1_s** in integration mode.
[0]	0	ATVALID	Read-only	Reads the value of **atvalid0_s** logically ORed with **atvalid1_s** in integration mode.

3.15 TPIU Integration Test FIFO Test Data Register 1, TPIU_ITFTTD1

This register indicates the integration status of the ATB data interfaces.

To read this register the integration data test mode must be enabled with TPIU_ITCTRL.Mode=`0b10`.

Reads are allowed otherwise but the register itself is disabled and doesn't get updated even if the inputs change.

Attributes

Offset

`0xEFC`

Type

Read-write

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

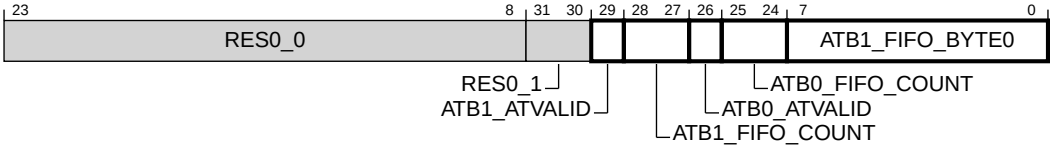

Figure 3-14 TPIU_ITFTTD1

The following table shows the register bit assignments.

Table 3-15 TPIU_ITFTTD1 attributes

Bits	Reset value	Name	Type	Function
[23:8]	0x0	RES0_0	Read-write	Reserved bit or field with SBZP behavior
[31:30]	0b00	RES0_1	Read-write	Reserved bit or field with SBZP behavior
[29]	0	ATB1_ATVALID	Read-only	Returns the value of the ATB1 Interface signal **atvalid1_s**. This field is RAZ if the TPIU-M does not include the ATB1 interface.
[28:27]	0b00	ATB1_FIFO_COUNT	Read-only	Number of bytes of ATB1 Interface trace data since last read of this register. This field is RAZ if the TPIU-M does not include the ATB1 interface. **0b00** 0 Bytes received and **atready1_s**=1 **0b01** 1 Byte received and **atready1_s**=0 **0b10** Reserved. **0b11** Reserved.
[26]	0	ATB0_ATVALID	Read-only	Returns the value of the ATB0 Interface signal **atvalid0_s**.

Table 3-15 TPIU_ITFTTD1 attributes (continued)

Bits	Reset value	Name	Type	Function
[25:24]	0b00	ATB0_FIFO_COUNT	Read-only	Number of bytes of ATB0 Interface trace data since last read of this register. **0b00** 0 Bytes received and **atready0_s**=1 **0b01** 1 Byte received and **atready0_s**=0 **0b10** Reserved. **0b11** Reserved.
[7:0]	0x0	ATB1_FIFO_BYTE0	Read-only	ATB1 Interface trace data FIFO byte 0. The TPIU-M discards this data when the register is read and as a consequence ATB1 FIFO byte count clears to `0x0`.

3.16 TPIU Integration Mode Control Register, TPIU_ITCTRL

This register is used to enable topology detection as described in CoreSight Architecture Specification. It enables TPIU-M to switch from functional mode (the default behaviour) to integration mode where the inputs and outputs of the component can be directly controlled for the purpose of integration testing and topology solving.

Integration Test Registers are provided to simplify the process of verifying the integration of the TPIU-M with other devices in a CoreSight system.These registers enable direct control of outputs and the ability to read the value of inputs. These registers must be used only when the TPIU_ITCTRL.Mode `0b00`.

Note: After entering integration test mode, the values of outputs controlled by the integration registers are UNKNOWN. They become valid only after the corresponding integration register has been appropriately written.

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

Attributes

Offset

`0xF00`

Type

Read-write

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

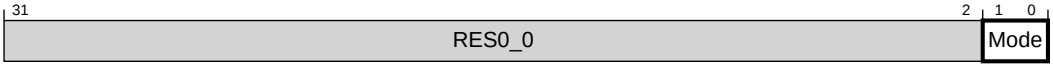

Figure 3-15 TPIU_ITCTRL

The following table shows the register bit assignments.

Table 3-16 TPIU_ITCTRL attributes

Bits	Reset value	Name	Type	Function
[31:2]	0x0	RES0_0	Read-write	Reserved bit or field with SBZP behavior
[1:0]	0b00	Mode	Read-write	Integration Mode Enable. **0b00** Normal mode. **0b01** Enable integration test mode. **0b10** Enable integration data test mode, which: • disables ATB0 or ATB1 traffic to reach formatter. • enables **atready0_s** generation to service 1 ATB transfer on the ATB0 interface. • enables **atready1_s** generation to service 1 ATB transfer on the ATB1 interface. • enables reading the data for serviced ATB transfers via TPIU_ITFTTD0 and TPIU_ITFTTD1. **0b11** Reserved.

3.17 TPIU Claim Tag Set Register, TPIU_CLAIMSET

This register forms one half of the claim tag value. On writes, this location enables individual bits to be set. On reads, it returns the number of bits that can be set.

Attributes

Offset

`0xFA0`

Type

Read-write

Reset

`0x0000000F`

Width

32

Bit descriptions

The following image shows the register bit assignments.

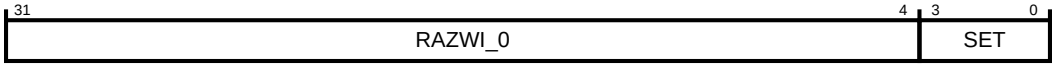

Figure 3-16 TPIU_CLAIMSET

The following table shows the register bit assignments.

Table 3-17 TPIU_CLAIMSET attributes

Bits	Reset value	Name	Type	Function
[31:4]	0x0	RAZWI_0	Read-write	RAZ/WI
[3:0]	0b1111	SET	Read-write	A bit-programmable register bank that sets the claim tag value. A read returns a logic 1 for all implemented locations.

3.18 TPIU Claim Tag Clear Register, TPIU_CLAIMCLR

This register forms one half of the claim tag value. On writes, this location enables individual bits to be cleared. On reads, it returns the current claim tag value.

Attributes

Offset

0xFA4

Type

Read-write

Reset

0x00000000

Width

32

Bit descriptions

The following image shows the register bit assignments.

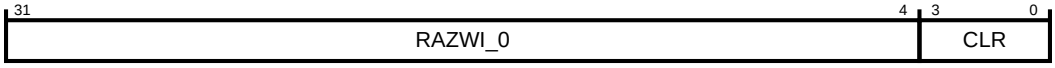

Figure 3-17 TPIU_CLAIMCLR

The following table shows the register bit assignments.

Table 3-18 TPIU_CLAIMCLR attributes

Bits	Reset value	Name	Type	Function
[31:4]	0x0	RAZWI_0	Read-write	RAZ/WI
[3:0]	0b0000	CLR	Read-write	A bit-programmable register bank that clears the claim tag value. It is zero at reset. It is used by software agents to signal to each other ownership of the hardware. It has no direct effect on the hardware itself.

3.19 TPIU Device Affinity register 0, TPIU_DEVAFF0

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

Attributes

Offset

`0xFA8`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

Figure 3-18 TPIU_DEVAFF0

The following table shows the register bit assignments.

Table 3-19 TPIU_DEVAFF0 attributes

Bits	Reset value	Name	Type	Function
[31:0]	0x0	DEVAFF0	Read-only	This field is RAZ.

3.20 TPIU Device Affinity register 1, TPIU_DEVAFF1

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

Attributes

Offset

`0xFAC`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

Figure 3-19 TPIU_DEVAFF1

The following table shows the register bit assignments.

Table 3-20 TPIU_DEVAFF1 attributes

Bits	Reset value	Name	Type	Function
[31:0]	0x0	DEVAFF1	Read-only	This field is RAZ.

3.21 TPIU Device Architecture Register, TPIU_DEVARCH

Identifies the architect and architecture of a CoreSight component. The architect might differ from the designer of a component, for example Arm defines the architecture but another company designs and implements the component.

Attributes

Offset

`0xFBC`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

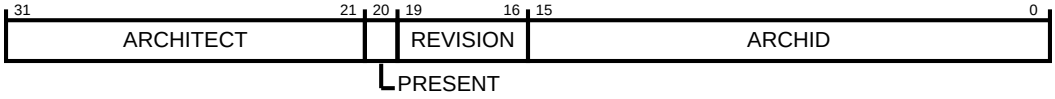

Figure 3-20 TPIU_DEVARCH

The following table shows the register bit assignments.

Table 3-21 TPIU_DEVARCH attributes

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

3.22 TPIU Device Configuration Register 2, TPIU_DEVID2

Contains an IMPLEMENTATION DEFINED value.

Attributes

Offset

`0xFC0`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

Figure 3-21 TPIU_DEVID2

The following table shows the register bit assignments.

Table 3-22 TPIU_DEVID2 attributes

Bits	Reset value	Name	Type	Function
[31:0]	0x0	DEVID2	Read-only	This field is RAZ.

3.23 TPIU Device Configuration Register 1, TPIU_DEVID1

Contains an IMPLEMENTATION DEFINED value.

Attributes

Offset

`0xFC4`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

Figure 3-22 TPIU_DEVID1

The following table shows the register bit assignments.

Table 3-23 TPIU_DEVID1 attributes

Bits	Reset value	Name	Type	Function
[31:0]	0x0	DEVID1	Read-only	This field is RAZ.

3.24 TPIU Device Identifier Register, TPIU_DEVID

Describes the TPIU to a debugger.

Attributes

Offset

0xFC8

Type

Read-only

Reset

0x00020C2-

Width

32

Bit descriptions

The following image shows the register bit assignments.

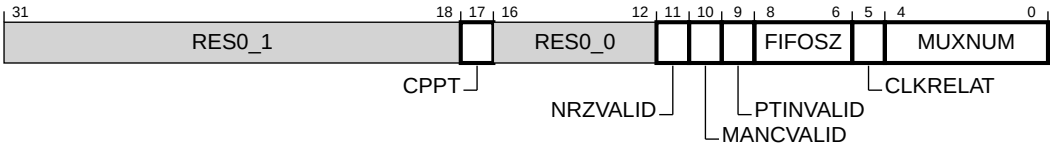

Figure 3-23 TPIU_DEVID

The following table shows the register bit assignments.

Table 3-24 TPIU_DEVID attributes

Bits	Reset value	Name	Type	Function
[31:18]	0x0	RES0_1	Read-only	Reserved bit or field with SBZP behavior
[17]	1	CPPT	Read-only	Clock Prescaler Parallel Trace. Indicates whether the Parallel trace port prescaler is controlled by TPIU_ACPR. This field is RES0 if TPIU_DEVID.PTINVALID == 1. **0** Parallel trace port is not affected by the prescaler controlled by TPIU_ACPR. **1** Parallel trace port is affected by the prescaler controlled by TPIU_ACPR.
[16:12]	0b00000	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[11]	1	NRZVALID	Read-only	NRZ valid. Indicates support for SWO using UART/NRZ encoding. **0** Not supported. **1** Supported.
[10]	1	MANCVALID	Read-only	Manchester valid. Indicates support for SWO using Manchester encoding. **0** Not supported. **1** Supported.

Table 3-24 TPIU_DEVID attributes (continued)

Bits	Reset value	Name	Type	Function
[9]	0	PTINVALID	Read-only	Trace Clock Plus Data support. Reads `0x0`, which indicates that trace clock and data is supported. **0** Supported. **1** Not supported.
[8:6]	0b000	FIFOSZ	Read-only	FIFO depth. Indicates the minimum implemented size of the TPIU output FIFO for trace data. The possible values of this field are: **0b000** IMPLEMENTATION DEFINED FIFO depth. **Other** Minimum FIFO size is 2^FIFOSZ. For example, a value of 0b011 indicates a FIFO size of at least 23 = 8 bytes.
[5]	1	CLKRELAT	Read-only	Relationship between clk and traceclkin. **0** Clocks are synchronous. **1** Clocks may be asynchronous.
[4:0]	0b00000	MUXNUM	Read-only	Indicates a hidden level of input multiplexing. **0** One ATB interface (ATB0 only, no input multiplexing). **1** Two ATB interfaces (ATB0 + ATB1).

3.25 TPIU Device Type Register, TPIU_DEVTYPE

Provides CoreSight Unique Component Identifier information for the TPIU.

Attributes

Offset

`0xFCC`

Type

Read-only

Reset

`0x00000011`

Width

32

Bit descriptions

The following image shows the register bit assignments.

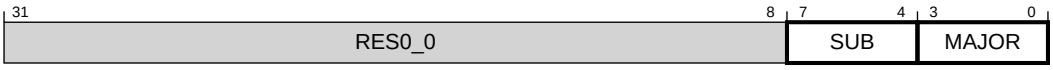

Figure 3-24 TPIU_DEVTYPE

The following table shows the register bit assignments.

Table 3-25 TPIU_DEVTYPE attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:4]	0b0001	SUB	Read-only	Sub-type. **0** Other **1** Trace Port
[3:0]	0b0001	MAJOR	Read-only	Major type. **0** Miscellaneous **1** Trace sink

3.26 TPIU Peripheral Identification Register 4, TPIU_PIDR4

The PIDR4 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFD0`

Type

Read-only

Reset

`0x00000004`

Width

32

Bit descriptions

The following image shows the register bit assignments.

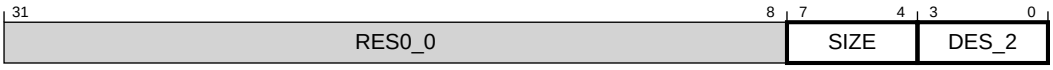

Figure 3-25 TPIU_PIDR4

The following table shows the register bit assignments.

Table 3-26 TPIU_PIDR4 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:4]	0b0000	SIZE	Read-only	Indicates the memory size that is used by this component. Returns 0 indicating that the component uses an UNKNOWN number of 4KB blocks. Using the SIZE field to indicate the size of the component is deprecated. Returns 0xB indicating Arm as the designer.
[3:0]	0b0100	DES_2	Read-only	JEP106 continuation code. Together, with PIDR2.DES_1 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same. Returns 0x4 indicating Arm as the designer.

3.27 TPIU Peripheral Identification Register 5, TPIU_PIDR5

The PIDR5 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFD4`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

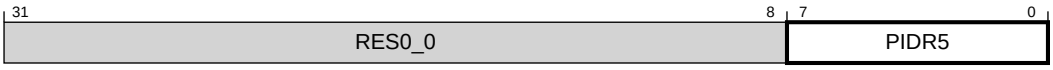

Figure 3-26 TPIU_PIDR5

The following table shows the register bit assignments.

Table 3-27 TPIU_PIDR5 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:0]	0x0	PIDR5	Read-only	Reserved.

3.28 TPIU Peripheral Identification Register 6, TPIU_PIDR6

The PIDR6 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFD8`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

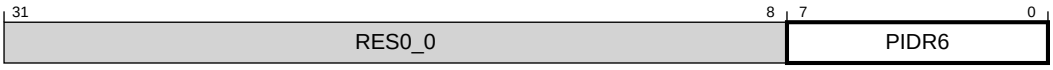

Figure 3-27 TPIU_PIDR6

The following table shows the register bit assignments.

Table 3-28 TPIU_PIDR6 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:0]	0x0	PIDR6	Read-only	Reserved.

3.29 TPIU Peripheral Identification Register 7, TPIU_PIDR7

The PIDR7 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFDC`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

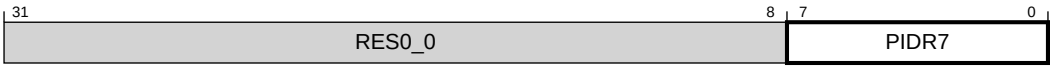

Figure 3-28 TPIU_PIDR7

The following table shows the register bit assignments.

Table 3-29 TPIU_PIDR7 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:0]	0x0	PIDR7	Read-only	Reserved.

3.30 TPIU Peripheral Identification Register 0, TPIU_PIDR0

The PIDR0 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFE0`

Type

Read-only

Reset

`0x000000F1`

Width

32

Bit descriptions

The following image shows the register bit assignments.

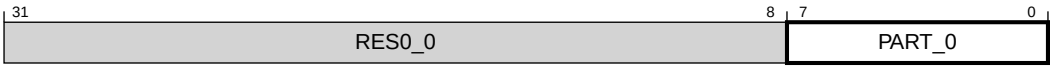

Figure 3-29 TPIU_PIDR0

The following table shows the register bit assignments.

Table 3-30 TPIU_PIDR0 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:0]	0xF1	PART_0	Read-only	Part number, bits[7:0]. Taken together with PIDR1.PART_1 it indicates the component. The Part Number is selected by the designer of the component.

3.31 TPIU Peripheral Identification Register 1, TPIU_PIDR1

The PIDR1 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFE4`

Type

Read-only

Reset

`0x000000B9`

Width

32

Bit descriptions

The following image shows the register bit assignments.

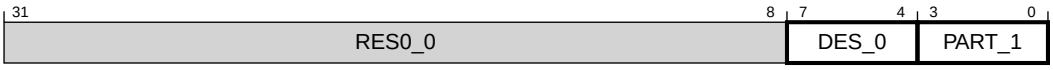

Figure 3-30 TPIU_PIDR1

The following table shows the register bit assignments.

Table 3-31 TPIU_PIDR1 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:4]	0b1011	DES_0	Read-only	JEP106 identification code, bits[3:0]. Together, with PIDR4.DES_2 and PIDR2.DES_1, they indicate the designer of the component and not the implementer, except where the two are the same.
[3:0]	0b1001	PART_1	Read-only	Part number, bits[11:8]. Taken together with PIDR0.PART_0 it indicates the component. The Part Number is selected by the designer of the component.

3.32 TPIU Peripheral Identification Register 2, TPIU_PIDR2

The PIDR2 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFE8`

Type

Read-only

Reset

`0x0000000B`

Width

32

Bit descriptions

The following image shows the register bit assignments.

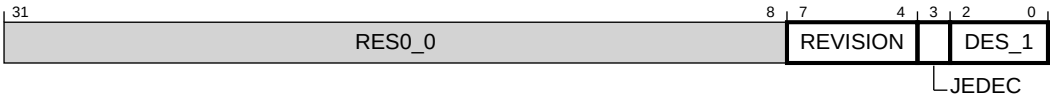

Figure 3-31 TPIU_PIDR2

The following table shows the register bit assignments.

Table 3-32 TPIU_PIDR2 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:4]	0b0000	REVISION	Read-only	Revision. It is an incremental value starting at `0x0` for the first design of a component. See *1.6 TPIU-M component* on page 1-17 for information on the RTL revision of the component.
[3]	1	JEDEC	Read-only	1 - Always set. Indicates that a JEDEC assigned value is used.
[2:0]	0b011	DES_1	Read-only	JEP106 identification code, bits[6:4]. Together, with PIDR4.DES_2 and PIDR1.DES_0, they indicate the designer of the component and not the implementer, except where the two are the same. Returns 0x3 indicating Arm as the designer.

3.33 TPIU Peripheral Identification Register 3, TPIU_PIDR3

The PIDR3 register is part of the set of peripheral identification registers.

Attributes

Offset

`0xFEC`

Type

Read-only

Reset

`0x00000000`

Width

32

Bit descriptions

The following image shows the register bit assignments.

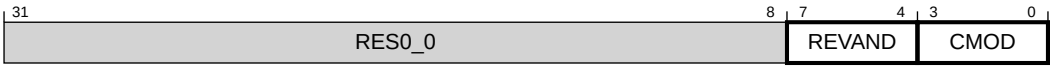

Figure 3-32 TPIU_PIDR3

The following table shows the register bit assignments.

Table 3-33 TPIU_PIDR3 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:4]	0b0000	REVAND	Read-only	This field indicates minor errata fixes specific to this design, for example metal fixes after implementation. In most cases this field is `0x0`.
[3:0]	0b0000	CMOD	Read-only	Customer Modified. Where the component is reusable IP, this value indicates if the customer has modified the behavior of the component. In most cases this field is `0x0`.

3.34 TPIU Component Identification Register 0, TPIU_CIDR0

The CIDR0 register is part of the set of component identification registers.

Attributes

Offset

`0xFF0`

Type

Read-only

Reset

`0x0000000D`

Width

32

Bit descriptions

The following image shows the register bit assignments.

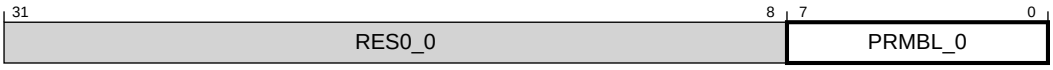

Figure 3-33 TPIU_CIDR0

The following table shows the register bit assignments.

Table 3-34 TPIU_CIDR0 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:0]	0xD	PRMBL_0	Read-only	Preamble. Returns `0x0D`.

3.35 TPIU Component Identification Register 1, TPIU_CIDR1

The CIDR1 register is part of the set of component identification registers.

Attributes

Offset

`0xFF4`

Type

Read-only

Reset

`0x00000090`

Width

32

Bit descriptions

The following image shows the register bit assignments.

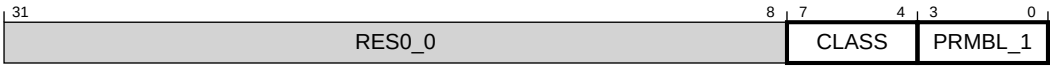

Figure 3-34 TPIU_CIDR1

The following table shows the register bit assignments.

Table 3-35 TPIU_CIDR1 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:4]	0b1001	CLASS	Read-only	Component class. Returns `0x9`, indicating this is a CoreSight component.
[3:0]	0b0000	PRMBL_1	Read-only	Preamble. Returns `0x0`.

3.36 TPIU Component Identification Register 2, TPIU_CIDR2

The CIDR2 register is part of the set of component identification registers.

Attributes

Offset

`0xFF8`

Type

Read-only

Reset

`0x00000005`

Width

32

Bit descriptions

The following image shows the register bit assignments.

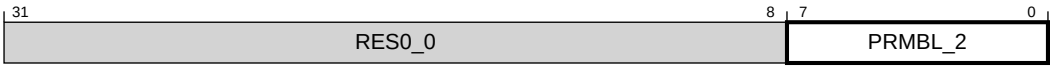

Figure 3-35 TPIU_CIDR2

The following table shows the register bit assignments.

Table 3-36 TPIU_CIDR2 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:0]	0x5	PRMBL_2	Read-only	Preamble. Returns `0x05`.

3.37 TPIU Component Identification Register 3, TPIU_CIDR3

The CIDR3 register is part of the set of component identification registers.

Attributes

Offset

`0xFFC`

Type

Read-only

Reset

`0x000000B1`

Width

32

Bit descriptions

The following image shows the register bit assignments.

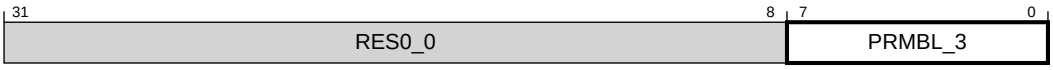

Figure 3-36 TPIU_CIDR3

The following table shows the register bit assignments.

Table 3-37 TPIU_CIDR3 attributes

Bits	Reset value	Name	Type	Function
[31:8]	0x0	RES0_0	Read-only	Reserved bit or field with SBZP behavior
[7:0]	0xB1	PRMBL_3	Read-only	Preamble. Returns `0xB1`.

Appendix A
Revisions

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

It contains the following section:

A.1 Revisions

Each table shows the technical differences between successive issues of the document.

Table A-1 Issue 0000-01

Change	Location
First release	-