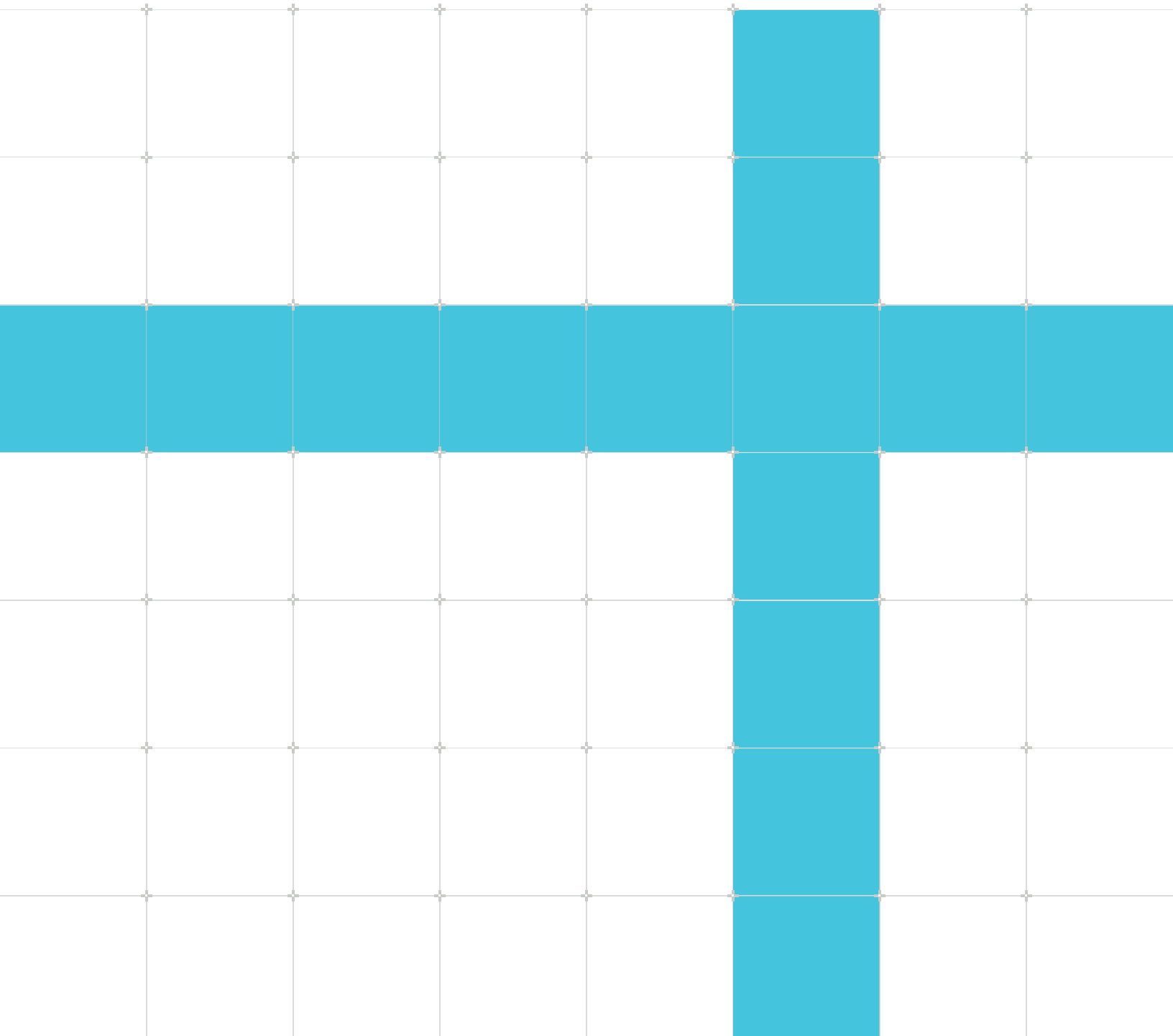
Arm® Corstone™-500

Revision: r0p0

Getting Started Guide

Issue 01
102262_0000_01_en



Release Information

Issue	Date	Confidentiality	Change
0000-01	12 September 2020	Non-Confidential	Initial release for r0p0.

Proprietary Notice

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. 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.

Web Address

developer.arm.com

Contents

1 Introduction

1.1 Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this manual, for example, r*1*p*2*, where:

r*m*	Identifies the major revision of the product, for example, r1.
p*n*	Identifies the minor revision or modification status of the product, for example, p2.

1.2 Intended audience

The Getting Started Guide is written for experienced hardware and System-on-Chip (SoC) engineers who might or might not have experience with Arm products. Such engineers typically have experience in writing Verilog and of performing synthesis, but might have limited experience of integrating and implementing Arm products.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

1.3.1 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: developer.arm.com/glossary.

1.3.2 Typographic conventions

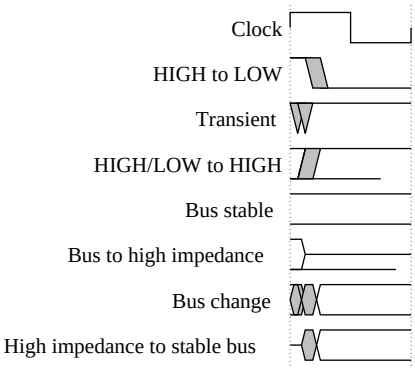
Convention	Use
italic	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace bold`	Denotes language keywords when used outside example code.
`monospace underline`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<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**.
	This represents a recommendation which, if not followed, might lead to system failure or damage.
	This represents a requirement for the system that, if not followed, might result in system failure or damage.
	This represents a requirement for the system that, if not followed, will result in system failure or damage.
	This represents an important piece of information that needs your attention.
	This represents a useful tip that might make it easier, better or faster to perform a task.
	This is a reminder of something important that relates to the information you are reading.

1.3.3 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



1.3.4 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.

1.4 Additional reading

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

Table 1: Arm publications

Document Name	Document ID	Licensee only
Arm® Corstone™-500 Getting Started Guide	102262	No
Arm® Corstone™-500 Reference Manual	102263	Yes
Arm® Corstone™-500 Release Note	PJDOC-1779577084-32989	Yes
Arm® Architecture Reference Manual Armv7-A and Armv7-R edition	DDI 0406C.D	No
Arm® MPS3 FPGA Prototyping Board Technical Reference Manual	100765	No
Arm® Fast Models Reference Manual v11.8	100964_1108_00_en	No
Arm® CoreSight™ SoC-400 Technical Reference Manual	100536	No
PrimeCell UART (PL011) Technical Reference Manual	DDI 0183	No
Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual	DDI 0224	No
Arm® True Random Number Generator (TRNG) Technical Reference Manual	100976	No
Arm® PrimeCell General Purpose Input/Output (PL061) Technical Reference Manual	DDI 0190	No
Arm® PrimeCell Synchronous Serial Port (PL022) Technical Reference Manual	DDI 0194	No
Arm® Server Base System Architecture 6.0 Platform Design Document	DEN 0029C	No

1.5 Feedback

Arm welcomes feedback on this product and its documentation.

1.5.1 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.

1.5.2 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® Corstone™-500 Getting Started Guide.
- The number 102262_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.

2 Overview of Corstone-500

Corstone-500 is an ideal starting point for feature rich System on Chip (SoC) designs based on the Cortex®-A5 core. These designs can be used in Internet of Things (IoT) and embedded products.

Corstone-500 includes a hardware and software package to enable you to develop your own subsystem to integrate into your custom SoC designs for high-performance Linux-capable silicon.

The key deliverable in Corstone-500 is the Cortex-A5 based Corstone SSE-500 Example Subsystem.

This full version of the product is fully expandable and configurable. You can customize SSE-500, the Field Programmable Gate Array (FPGA), and the included IP to meet your design requirements.

2.1 Creating a SoC with Corstone-500

Corstone-500 can be used in your design flow in a number of ways:

- To become familiar with Arm IP and flows.
- To demonstrate a proof of concept.
- To prototype peripherals and accelerators.
- To model the design of a custom chip.
- To analyze the performance of a system design.

Simulation using Corstone-500 is a first step in a wide range of possible design flows that can lead you to develop your own products based on Arm technology.

2.2 What is included in Corstone-500

Corstone-500 includes most of the Arm IP in the SSE-500 subsystem and example integration layer, an FPGA, and access to modelling options.


Note

The Cortex-A5 core and CoreSight ETM-A5 Embedded Trace Macrocell for Cortex-A5 are not included in Corstone-500 and need to be licensed and downloaded separately.

Corstone-500 includes:

Arm Corstone SSE-500 Example Subsystem

- The SSE-500 subsystem is a Cortex-A5 based subsystem that incorporates Arm CoreLink™ NIC-400 Network Interconnect and Arm CoreSight™ SoC-400 components, and other Arm IP.
- The SSE-500 example integration layer gives an example of how to integrate the subsystem into a full SoC design.
- See Arm Corstone SSE-500 Example Subsystem and *Arm® Corstone™-500 Reference Manual*.

Arm IP

- Bundles of Cortex-A5 (available separately) and System IP for designing and implementing a SoC based on the SSE-500 design or using the included IP.
- See Subsystem IP and components and Example integration layer IP and components.

FPGAs for prototyping

- An example FPGA image that you can use for prototyping and debugging on the Arm MPS3 FPGA Prototyping Board.
- Guidance for creating your own FPGA image based on SSE-500.
- See Prototyping with FPGA images and the MPS3 board, and *Arm® Corstone™-500 Reference Manual* for information on installing, rendering, and working with the FPGA for Corstone-500.

Simulation models

- A Fast Model and an integrated software stack, so that you can start to develop Linux-based software before your hardware is available.
- See Modeling your Corstone-500 system.

2.3 The Corstone-500 documents

Arm delivers a documentation set with its products.

The documents for Corstone-500 are:

- *Arm® Corstone™-500 Release Note*

  The Release Note describes the deliverables in the Corstone-500 bundle and gives information regarding the specific release.

- *Arm® Corstone™-500 Getting Started Guide*

  The Getting Started Guide is a non-confidential document that gives an overview of Corstone-500. The guide describes the IP in SSE-500, the prototyping and modelling options, and programming information for the subsystem. Read this guide if you are new to Arm IP or

want an overview of how Corstone-500 can be used as the starting point for creating your own Internet of Things (IoT) application.

- *Arm® Corstone™-500 Reference Manual*

  The Reference Manual describes the functionality of SSE-500 and FPGA with appropriate programming information. Configuration, integration, and implementation details are given to enable you to create your own custom chip.

The other documents for Arm IP referred to in this guide are in their respective IP bundles.

3 Arm Corstone SSE-500 Example Subsystem

The subsystem and example integration layer are the starting point for your design. The topology gives a top-level overview of SSE-500 and the FPGA.

- The SSE-500 subsystem implements the major functionality of a system that is able to support embedded Linux.
- The example integration layer gives you an example of how to integrate the subsystem into your custom SoC.
- The FPGA is an implementation of SSE-500 that can be used for prototyping on an Arm MPS3 FPGA Prototyping Board. For more details, see Prototyping with FPGA images and the MPS3 board.

For more details, see the Reference Manual.

3.1 SSE-500 topology

The subsystem is designed using Arm IP and components, and is partitioned into the following functional blocks:

- Processor block
- Debug block
- Interconnect block
- Peripheral block
- Clock and reset block

The following figure shows the IP, components, and connections in SSE-500.

Figure 2: SSE-500 topology

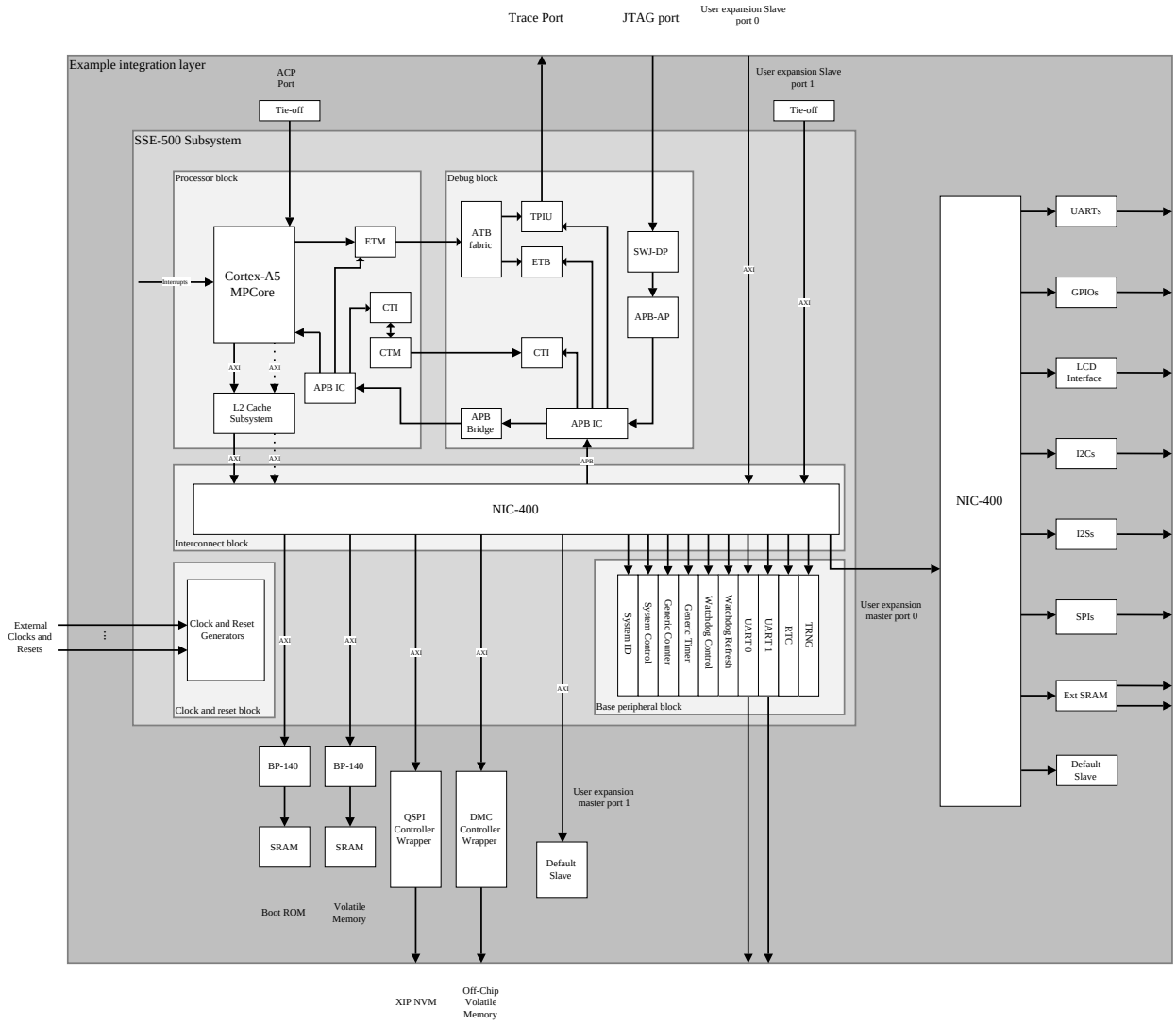


Table 2: Abbreviations used in SSE-500 topology

Abbreviation	Description
APB IC	Advanced Peripheral Bus Interconnect
APB-AP	Part of the Debug Access Port that provides an APB interface to a SoC for the debugger
ATB	An AMBA® bus protocol for trace data
BP-140	PrimeCell Infrastructure AMBA 3 AXI Internal Memory Interface
CTI	Cross Trigger Interface
CTM	Cross Trigger Matrix
DMC	Dynamic Memory Controller
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
I2C	Inter-Integrated Circuit controller
I2S	Inter-IC Sound Bus Controller
QSPI	Quad Serial Port Interface
RTC	Real Time Clock
SPIs	Serial Peripheral Interface
SWJ-DP	Combined JTAG-DP and SW-DP
TPIU	Trace Port Interface Unit
TRNG	True Random Number Generator

3.2 Subsystem IP and components

The subsystem consists of the following Arm IP:

Cortex-A5 MPCore processor with optional FPU and Neon™ extensions [1]

The Cortex-A5 MPCore processor is a high-performance, low-power, Arm macrocell with an L1 cache subsystem that provides full virtual memory capabilities. Up to four individual cores

can be linked in a cache-coherent cluster. The cluster is under the control of a Snoop Control Unit (SCU), that maintains L1 data cache coherency for memory marked as shared.

CoreLink Level 2 Cache Controller L2C-310

The addition of an on-chip secondary cache, also referred to as a Level 2 or L2 cache, is a recognized method of improving the performance of Arm-based systems when the processor generates significant memory traffic. The cache controller is a unified, physically addressed, physically tagged cache with up to 16 ways.

CoreSight SoC-400 solution for debug and trace

CoreSight SoC-400 is a comprehensive library of components for the creation of debug and trace functionality within a system.

CoreSight ETM-A5 Embedded Trace Macrocell for Cortex-A5 [1]

The ETM-A5 is a real-time trace module providing instruction and data tracing of the Cortex-A5 microprocessor.

CoreLink NIC-400 Network Interconnect

The NIC-400 enables you to create a complete high performance, optimized, and AMBA®-compliant network infrastructure that can range from a single bridge component to a complex interconnect of up to 128 masters and 64 slaves of AMBA protocols.

True Random Number Generator (CC003)

The True Random Number Generator (TRNG) provides an assured level of entropy (as analyzed by Entropy Estimation logic). You can use the output from the TRNG to seed deterministic random bit generators.

PrimeCell Real Time Clock (PL031)

The Real Time Clock (RTC) is an AMBA slave module that connects to the Advanced Peripheral Bus (APB). A 1Hz clock input to the RTC generates counting in one second intervals. The RTC provides an alarm function or long time base counter by generating an interrupt signal after counting a programmed number of cycles of the clock input.

PrimeCell UART (PL011)

The UART is an Advanced Microcontroller Bus Architecture (AMBA) compliant SoC peripheral that is developed, tested, and licensed by Arm. The UART is an AMBA slave module that connects to the Advanced Peripheral Bus (APB). The UART includes an Infrared Data Association (IrDA) Serial InfraRed (SIR) protocol ENcoder/DECoder (ENDEC).

PrimeCell General Purpose Input/Output (PL061)

The PrimeCell GPIO is an Advanced Microcontroller Bus Architecture (AMBA) bus slave that connects to the AMBA Advanced Peripheral Bus (APB). It provides eight programmable inputs or outputs that you can control in either software or hardware modes.

PrimeCell Synchronous Serial Port (PL022)

The PrimeCell Synchronous Serial Port (SSP) is an Advanced Microcontroller Bus Architecture (AMBA) slave block that connects to the Advanced Peripheral Bus (APB). The PrimeCell SSP is an AMBA compliant System-on-Chip (SoC) peripheral that is developed, tested, and licensed by Arm.

Generic timer and counter

The subsystem includes a Generic Timer and Generic Counter as specified in the Arm ARM. See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition* Section B8 and Appendix E for the programmers model for Generic Timers and Counter.

Generic watchdog

If the watchdog is not refreshed within a certain time it raises an interrupt signal. The watchdog uses the Generic Timer system counter as the time base against which the decision to trigger an interrupt is made.


Note

1. Cortex-A5 and ETM-A5 are not included in the Corstone-500 bundle and must be obtained separately.

3.3 Example integration layer IP and components

The example integration layer shows how additional IP and components can be integrated around the subsystem to form a complete system or SoC.

The example integration layer includes:

- Static Memory Controller (SMC) from PrimeCell Infrastructure AMBA® 3 AXI™ Internal Memory Interface (BP140) (non-programmable component)
- ROM and RAM.
- External interconnect, that is implemented by NIC-400
- Quad Serial Port Interface (QSPI) controller wrapper which, depending on your requirements, can be replaced with a third-party QSPI controller
- Dynamic Memory Controller (DMC) wrapper which, depending on your requirements, can be replaced with a third-party DMC
- External peripherals:
  - Inter-Integrated Circuit controller (I2C)
  - Inter-IC Sound Bus Controller (I2S)
  - PrimeCell GPIO (PL061) general purpose I/O device
  - PrimeCell UART (PL011)
  - Color LCD (CLCD) interface
  - SPI from PrimeCell Synchronous Serial Port (PL022)

4 Prototyping with FPGA images and the MPS3 board

The Arm MPS3 FPGA Prototyping Board is an FPGA IoT development platform. The board is designed to support small to medium Arm Cortex-A processors, or dedicated custom designs.

Corstone-500 includes a preconfigured FPGA image that you can load onto the MPS3 board. Alternatively, if you make changes to the default SSE-500 configuration, you can create and load your own FPGA image. You can load images onto the board using a microSD card.

The MPS3 board enables you to connect sensors, motors, and other design-specific peripherals.

You can also use the board to connect the FPGA to Arm Development Studio for testing and debugging of your design.

For more details:

- See *Arm® Corstone™-500 Reference Manual* for information on installing, rendering and working with the FPGA for Corstone-500.
- See *Arm® MPS3 FPGA Prototyping Board Technical Reference Manual* for details about the MPS3 board and its usage.
- Visit https://developer.arm.com/tools-and-software/development-boards/fpga-prototyping-boards/mps3 for the latest information and to buy the MPS3 board.

5 Modeling your Corstone-500 system

There are different models that are available for designing and testing a system. These models enable you to develop software ahead of hardware availability.

For more details, and to download models and software for Corstone-500, visit https://developer.arm.com/tools-and-software/corstone-500-models.

5.1 Models

Corstone-500 includes an Arm Fast Model.

Fast Model

Arm Fast Models are fast, functionally accurate models suited to software development or high-level modeling. The Corstone-500 Fast Model is delivered as a pre-built Fixed Virtual Platform (FVP) which is quick and easy to setup and start your software development. The FVP works with the software stack to give out-of-box Linux support, see Software.

The model is provided with a high level of instrumentation and supports debug access by source level debuggers such as Arm Development Studio and others from Arm ecosystem partners. Tools for customizing the model can be purchased separately.

All models are based on the Arm Versatile™ Express memory map. For a full description of the VE memory map for Cortex-A series, see the *Arm® Fast Models Reference Manual v11.8*.

5.2 Software

An extensible software stack is available that provides a reference Linux Operating System to accelerate your application on Corstone-500.

The software stack represents a fully integrated and operational model of the Corstone-500 system.

The reference stack can be modified and configured to meet your specific requirements.

Appendix A Programmers model

The Programmers Model describes the SSE-500 subsystem and integration layer memory regions and registers. This information is needed when you program a SoC that contains an implementation of the subsystem.

To use this information to program a model you do not have to have a Corstone-500 license.

The following information applies to all registers:

- The access type of a register is either:
  - Read and write (RW)
  - Read-only (RO)
  - Write-only (WO)
  - Read-As-Zero and Write-Ignored (**RAZ/WI**)
- Reserved locations follow these guidelines:
  - If a peripheral or device does not occupy all the memory allocated to it, the unoccupied region is Reserved.
  - Locations which are marked as Reserved are treated as **RAZ/WI** and generate an error when accessed.
  - Attempting to access Reserved or unused address locations can result in unpredictable behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify undefined register bits.
  - Ignore undefined register bits on reads.
  - Register bits are reset to a logic 0 by a system or power on reset.

A.1 Memory map

All peripherals and devices in the subsystem are allocated memory units of at least 64KB to allow the use of any page granule size defined by *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*.

A.1.1 Subsystem memory map

The subsystem memory map describes the base address and region information for the subsystem.

Table 3: Subsystem memory map

Base address	Size	Component	Notes
0x0000_0000	64KB	SRAM0 for ROM	Default implementation region
0x0001_0000	960KB	SRAM0 for ROM	Extendable region, any unmapped section of this region is aliased
0x0010_0000	15MB	Reserved	-
0x0100_0000	16MB	Reserved	-
0x0200_0000	2MB	SRAM1	Default implementation region
0x0220_0000	94MB	SRAM1	Extendable region, any unmapped section of this region is aliased
0x0800_0000	8MB	QSPI flash	Default implementation region
0x0880_0000	120MB	QSPI flash	Extendable region, any unmapped section of this region is aliased
0x1000_0000	160MB	Debug	See Self-hosted debug memory map
0x1A00_0000	32MB	Base peripherals	See Base peripherals memory map
0x1C00_0000	64KB	A5 Peripherals	See *Arm® Cortex®-A5 MPCore Technical Reference Manual*
0x1C01_0000	64KB	L2C-310 Peripherals	See *Arm® CoreLink™ Level 2 Cache Controller L2C-310 Technical Reference Manual* **Note:** Since the presence of L2C-310 is configurable, this region is Reserved if L2C-310 is not present.
0x1C02_0000	575MB	Reserved	-
0x4000_0000	512MB	Master expansion port 0	Used in the integration layer for connecting external peripherals. See Integration layer memory map.
0x6000_0000	512MB	Master expansion port 1	-
0x8000_0000	2GB	DDR	-

A.1.1.1 Self-hosted debug memory map

The self-hosted debug memory map describes the offset of the subsystem debug components relative to the Debug region base address (`0x1000_0000`).

Table 4: Self-hosted debug memory map

Offset	Size	Component	Notes
`0x00_0000`	64KB	System Debug ROM	See *Arm® CoreSight™ SoC-400 Technical Reference Manual*
`0x01_0000`	64KB	Replicator	-
`0x02_0000`	64KB	TPIU	-
`0x03_0000`	64KB	ETB	-
`0x04_0000`	64KB	System CTI	-
`0x05_0000`	64KB	Timestamp Generator	-
`0x06_0000`	640KB	Reserved	-
`0x10_0000`	128KB	CPU Debug	See CPU debug memory map
`0x12_0000`	896KB	Reserved	-
`0x20_0000`	158MB	Reserved	-

A.1.1.1.1 CPU debug memory map

The CPU debug memory map lists the address offset of the CPU debug components relative to the CPU Debug region base address in:

- Self-hosted debug memory map (`0x1010_0000`).
- External debug memory map (`0x8010_0000`).

Table 5: CPU debug memory map

Offset	Size	Component	Notes
`0x0_0000`	4KB	Cortex-A5 PIL ROM table	-
`0x0_1000`	60KB	Reserved	-

Offset	Size	Component	Notes
0x1_0000	4KB	Processor0 Debug	-
0x1_1000	4KB	Processor0 PMU	-
0x1_2000	4KB	Processor1 Debug	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
0x1_3000	4KB	Processor1 PMU	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
0x1_4000	4KB	Processor2 Debug	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.
0x1_5000	4KB	Processor2 PMU	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.
0x1_6000	4KB	Processor3 Debug	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.
0x1_7000	4KB	Processor3 PMU	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.
0x1_8000	4KB	CTI0	-
0x1_9000	4KB	CTI1	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
0x1_A000	4KB	CTI2	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.
0x1_B000	4KB	CTI3	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.
0x1_C000	4KB	ETM0	-
0x1_D000	4KB	ETM1	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
0x1_E000	4KB	ETM2	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.
0x1_F000	4KB	ETM3	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.

A.1.1.2 Base peripherals memory map

The base peripherals memory map describes the base address and region information for the peripherals.

Table 6: Base peripherals memory map

Base address	Size	Component	Notes
0x1A00_0000	64KB	System ID	See System ID registers
0x1A01_0000	64KB	System control	See System control registers
0x1A02_0000	64KB	Generic counter control	See Generic counter control registers
0x1A03_0000	64KB	Generic counter read	See Generic counter read registers
0x1A04_0000	64KB	Generic timer control	See Generic timer control registers
0x1A05_0000	64KB	Generic timer 0	See Generic timer registers
0x1A06_0000	640KB	Reserved	-
0x1A10_0000	64KB	Watchdog control	See Generic watchdog control registers
0x1A11_0000	64KB	Watchdog refresh	See Generic watchdog refresh registers
0x1A12_0000	896KB	Reserved	-
0x1A20_0000	64KB	UART0	See *PrimeCell UART (PL011) Technical Reference Manual*
0x1A21_0000	64KB	UART1	See *PrimeCell UART (PL011) Technical Reference Manual*
0x1A22_0000	64KB	RTC	See *Arm® PrimeCell Real Time Clock (PL031) Technical Reference Manual*
0x1A23_0000	64KB	TRNG	See *Arm® True Random Number Generator (TRNG) Technical Reference Manual*
0x1A24_0000	29MB	Reserved	-

A.1.1.3 External debug memory map

When the debug subsystem is accessed by an external debugger, bit 31 of the APB address bus must be set high. This bit is used by the CoreSight SoC-400 components to distinguish between external debug access and self-hosted debug access. Therefore, the external debugger memory map starts at base address `0x8000_0000`.

Table 7: External Debug memory map

Base address	Size	Component	Notes
`0x8000_0000`	64KB	System Debug ROM	See *Arm® CoreSight™ SoC-400 Technical Reference Manual*
`0x8001_0000`	64KB	Replicator	-
`0x8002_0000`	64KB	TPIU	-
`0x8003_0000`	64KB	ETB	-
`0x8004_0000`	64KB	System CTI	-
`0x8005_0000`	64KB	Timestamp Generator	-
`0x8006_0000`	640KB	Reserved	-
`0x8010_0000`	128KB	CPU Debug	See CPU debug memory map
`0x8012_0000`	896KB	Reserved	-

A.1.2 Integration layer memory map

The integration layer uses one expansion master port to connect to the NIC-400. The integration layer memory map contains memory information for all the peripherals in the example integration layer.

Table 8: Integration Layer Memory Map

Base address	Size	Component	Notes
`0x4000_0000`	64KB	UART2	See *PrimeCell UART (PL011) Technical Reference Manual*
`0x4001_0000`	64KB	UART3	-
`0x4002_0000`	64KB	UART4	-

Base address	Size	Component	Notes
0x4003_0000	64KB	UART5	-
0x4004_0000	64KB	GPIO0	See *Arm® PrimeCell General Purpose Input/Output (PL061) Technical Reference Manual*
0x4005_0000	64KB	GPIO1	-
0x4006_0000	64KB	GPIO2	-
0x4007_0000	64KB	GPIO3	-
0x4008_0000	64KB	GPIO4	-
0x4009_0000	64KB	GPIO5	-
0x400A_0000	64KB	GPIO6	-
0x400B_0000	64KB	GPIO7	-
0x400C_0000	256KB	Reserved	-
0x4010_0000	64KB	I2S0	See I2S
0x4011_0000	64KB	I2S1	-
0x4012_0000	64KB	I2C0	See I2C
0x4013_0000	64KB	I2C1	-
0x4014_0000	64KB	I2C2	-
0x4015_0000	64KB	I2C3	-
0x4016_0000	64KB	I2C4	-
0x4017_0000	64KB	SPI0	See *Arm® PrimeCell Synchronous Serial Port (PL022) Technical Reference Manual*
0x4018_0000	64KB	SPI1	-
0x4019_0000	64KB	SPI2	-
0x401A_0000	64KB	CLCD Config Reg	See CLCD Interface registers
0x401B_0000	320KB	Reserved	-

Base address	Size	Component	Notes
0x4020_0000	1MB	Ethernet	-
0x4030_0000	1MB	USB	-
0x4040_0000	508MB	USEREXT0	Expansion space on integration layer interconnect connected to a default slave

A.2 Register Descriptions

The registers described in this section complement those described in the *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*. See also the technical documentation of the included Arm IP and any model you want to use.

A.2.1 Subsystem registers

The subsystem registers provide information needed to program the subsystem. The registers including system ID, system control, and generic counter, timer, and watchdog resisters.

A.2.1.1 System ID registers

System ID registers describe system configuration and identification information.

Table 9: System ID register summary

Offset	Name	Type	Reset	Width	Description
0x000	Reserved	**RAZ/WI**	-	-	Reserved
0x004	Reserved	**RAZ/WI**	-	-	Reserved
0x008	SYS_CFG2	RO	0x0022_0000	32-bit	System configuration 2
0x040	SOC_ID	RO	CFG_DEF	32-bit	SoC identification
0xFC8	IIDR	RO	0x7F00043B	32-bit	Implementer identification register
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4

Offset	Name	Type	Reset	Width	Description
0xFD4	Reserved	**RAZ/WI**	-	-	-
0xFD8	Reserved	**RAZ/WI**	-	-	-
0xFDC	Reserved	**RAZ/WI**	-	-	-
0xFD4	Reserved	**RAZ/WI**	-	-	-
0xFE0	CNTPIDR0	RO	0xB0	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	0xB8	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	0x0B	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3
0xFF0	CNTCIDR0	RO	0x0D	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.1.1 SYS_CFG2, System Configuration 2

SYS_CFG2 contains read-only information about the number of expansion master interfaces and slave interfaces.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

0x0000_0008

Bit descriptions

Table 10: SYS_CFG2 bit descriptions

Bits	Types	Reset	Name	Description
31:24	RO	0x00	RESERVED	Reserved
23:20	RO	0x2	NUM_EXP_MST	Number of expansion master interface, 0x2 = 2 expansion master interfaces. All other values are reserved.
19:16	RO	0x2	NUM_EXP_SLV	Number of expansion slave interface, 0x2 = 2 expansion slave interfaces. All other values are reserved.
15:0	RO	0x0000	RESERVED	Reserved

A.2.1.1.2 SOC_ID, SoC identification

SOC_ID contains information to identify a SoC that has been derived from the Corstone-500 subsystem, and the implementer of the SoC. The value of this register is driven by incoming signals that are external to the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

0x0000_0040

Bit descriptions

Table 11: SOC_ID bit descriptions

Bits	Types	Reset	Name	Description
31:20	RO	CFG_DEF	PRODUCT_ID	SoC product identification.

Bits	Types	Reset	Name	Description
19:16	RO	CFG_DEF	VARIANT	Variant of SoC.
15:12	RO	CFG_DEF	REVISION	Minor Revision of SoC.
11:0	RO	CFG_DEF	IMPLEMENTER	JEP106 Code of company implementing SoC

A.2.1.1.3 IIDR, Implementer identification

IIDR contains information to identify the implementer of the subsystem architecture.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

`0x0000_0FC8`

Bit descriptions

Table 12: IIDR bit descriptions

Bits	Types	Reset	Name	Description
31:20	RO	`0x7F0`	PRODUCT_ID	Subsystem product identification
19:16	RO	`0x0`	VARIANT	Variant of the subsystem
15:12	RO	`0x0`	REVISION	Revision of the subsystem
11:0	RO	`0x43B`	IMPLEMENTER	JEP106 Code of company implementing the subsystem: • [11:8] JEP106 Continuation Code • [7] Always 0 • [6:0] JEP106 identify code

A.2.1.2 System control registers

The system control registers describe control information in the Corstone-500 subsystem.

The following table lists the system control registers.

Table 13: System control register summary

Offset	Name	Type	Reset	Width	Description
0x000	RESERVED	RW	-	32-bit	Reserved
0x010	PE0_CONFIG	RW	0x0000_0000	32-bit	Processing Element 0 Static Config
0x020	PE1_CONFIG	RW	0x0000_0000	32-bit	Processing Element 1 Static Config
0x030	PE2_CONFIG	RW	0x0000_0000	32-bit	Processing Element 2 Static Config
0x040	PE3_CONFIG	RW	0x0000_0000	32-bit	Processing Element 3 Static Config
0x200	SYS_RST_SYN	RO	See individual bit resets	32-bit	System reset syndrome
0x330	SYS_RST_CTL	WO	0x0000_0000	32-bit	System reset control
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4
0xFD4	Reserved	**RAZ/WI**	-	-	-
0xFD8	Reserved	**RAZ/WI**	-	-	-
0xFDC	Reserved	**RAZ/WI**	-	-	-
0xFD4	Reserved	**RAZ/WI**	-	-	-
0xFE0	CNTPIDR0	RO	0xB1	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	0xB8	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	0x0B	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3

Offset	Name	Type	Reset	Width	Description
0xFF0	CNTCIDR0	RO	0x0D	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.2.1 PE0_CONFIG, Processing Element 0 Static Config

PE0_CONFIG is used to control the configuration options of the Processing Element or core 0 of the processor cluster.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System control registers

Address offset

0x0000_0010

Bit descriptions

Table 14: PE0_CONFIG bit descriptions

Bits	Types	Reset	Name	Description
31:3	RO	0x000_0000	RESERVED	Reserved
2	RW	0	VINITHI	Location of the exception vectors at reset: • 0 - Vector starts at 0x0000_0000 • 1 - Vector starts at 0xFFFF_0000

Bits	Types	Reset	Name	Description
1	RW	0	TEINIT	Enabling Thumb® exceptions: • 0 - Exceptions taken in Arm(v7) state • 1 - Exceptions taken in Thumb state
0	RW	0	CFGEND	Endianness configuration: • 0 - Little endian • 1 - Big endian

A.2.1.2.2 PE1_CONFIG, Processing Element 1 Static Config

PE1_CONFIG is used to control the configuration options of the Processing Element or core 1 of the processor cluster.

Configurations

This register is available in configurations with more than one core.

Attributes

Width

32-bit

Functional group

System control registers

Address offset

`0x0000_0020`

Bit descriptions

Table 15: PE1_CONFIG bit descriptions

Bits	Types	Reset	Name	Description
31:3	RO	`0x000_0000`	RESERVED	Reserved
2	RW	0	VINITHI	Location of the exception vectors at reset: • 0 - Vector starts at `0x0000_0000` • 1 - Vector starts at `0xFFFF_0000`

Bits	Types	Reset	Name	Description
1	RW	0	TEINIT	Enabling Thumb exceptions: • 0 - Exceptions taken in Arm(v7) state • 1 - Exceptions taken in Thumb state
0	RW	0	CFGEND	Endianness configuration: • 0 - Little endian • 1 - Big endian

A.2.1.2.3 PE2_CONFIG, Processing Element 2 Static Config

PE2_CONFIG is used to control the configuration options of the Processing Element or core 2 of the processor cluster.

Configurations

This register is available in configurations with more than two cores.

Attributes

Width

32-bit

Functional group

System control registers

Address offset

`0x0000_0030`

Bit descriptions

Table 16: PE2_CONFIG bit descriptions

Bits	Types	Reset	Name	Description
31:3	RO	`0x000_0000`	RESERVED	Reserved
2	RW	0	VINITHI	Location of the exception vectors at reset: • 0 - Vector starts at `0x0000_0000` • 1 - Vector starts at `0xFFFF_0000`

Bits	Types	Reset	Name	Description
1	RW	0	TEINIT	Enabling Thumb exceptions: • 0 - Exceptions taken in Arm(v7) state • 1 - Exceptions taken in Thumb state
0	RW	0	CFGEND	Endianness configuration: • 0 - Little endian • 1 - Big endian

A.2.1.2.4 PE3_CONFIG, Processing Element 3 Static Config

Processing element static configuration register is used to control the configuration options of the Processing Element or core 3 of the processor cluster.

Configurations

This register is available in configurations with more than three cores.

Attributes

Width

32-bit

Functional group

System control registers

Address offset

`0x0000_0040`

Bit descriptions

Table 17: PE3_CONFIG bit descriptions

Bits	Types	Reset	Name	Description
31:3	RO	`0x000_0000`	RESERVED	Reserved
2	RW	0	VINITHI	Location of the exception vectors at reset: • 0 - Vector starts at `0x0000_0000` • 1 - Vector starts at `0xFFFF_0000`

Bits	Types	Reset	Name	Description
1	RW	0	TEINIT	Enabling Thumb exceptions: • 0 - Exceptions taken in Arm(v7) state • 1 - Exceptions taken in Thumb state
0	RW	0	CFGEND	Endianness configuration: • 0 - Little endian • 1 - Big endian

A.2.1.2.5 SYS_RST_SYN, System Reset Syndrome

SYS_RST_SYN stores information on the cause of most recent reset.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System control registers

Address offset

0x0000_0200

Bit descriptions

Table 18: SYS_RST_SYN bit descriptions

Bits	Types	Default	Name	Description
31:4	RO	0x000_0000	RESERVED	Reserved
3	RO	0	SYSRSTREQ	When HIGH, indicates that the last reset of system was caused by SW programmed SYSRSTREQ being asserted.
2	RO	0	nSRST	When HIGH, indicates that the last reset of system was caused by nSRST pin being asserted.

Bits	Types	Default	Name	Description
1	RO	0	SYSWDIRQ	When HIGH, indicates that the last reset of system was caused by system watchdog second expiry signal being asserted.
0	RO	1	PORESETn	Indicates that the last reset of system was caused by PORESETn pin being asserted

A.2.1.2.6 SYS_RST_CTL, System Reset Control

SYS_RST_CTL provides a software programmable mechanism to reset the system.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System control registers

Address offset

0x0000_0330

Bit descriptions

Table 19: SYS_RST_CTL bit descriptions

Bits	Types	Reset	Name	Description
31:2	RO	0x000_0000	RESERVED	Reserved
1	RW	0	RST_REQ	System reset request: • 0 - No reset requested • 1 - Reset requested
0	RO	0	RESERVED	Reserved

A.2.1.3 Generic counter control registers

Generic counter control registers provide the programming interface to the control frame of the generic counter.

Table 20: Generic counter control register summary

Offset	Name	Type	Reset	Width	Description
0x000	CNTCR (Counter Control Register)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x004	CNTSR (Counter Status Register)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x008	CNTCV[31:0] (Counter count value low)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x00C	CNTCV[63:32] (Counter count value high)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x010 - 0x01C	Reserved	**RAZ/WI**	-	-	-
0x020	CNTFID0 (Base Frequency)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x024	Frequency Mode Table Endmark	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x028 - 0xFCC	Reserved	**RAZ/WI**	-	-	-
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	**RAZ/WI**	-	-	-
0xFE0	CNTPIDR0	RO	0x9C	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	0xB0	32-bit	Peripheral ID 1

Offset	Name	Type	Reset	Width	Description
0xFE8	CNTPIDR2	RO	0x0B	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3
0xFF0	CNTCIDR0	RO	0x0D	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.4 Generic counter read registers

Generic counter read registers provide the programming interface to the read frame of the generic counter.

Table 21: Generic counter read register summary

Offset	Name	Type	Reset	Width	Description
0x000	CNTCV[31:0] (Counter count value low)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x004	CNTCV[63:32] (Counter count value high)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x008 - 0xFCC	Reserved	**RAZ/WI**	-	-	-
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	**RAZ/WI**	-	-	-
0xFE0	CNTPIDR0	RO	0x9D	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	0xB0	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	0x0B	32-bit	Peripheral ID 2

Offset	Name	Type	Reset	Width	Description
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3
0xFF0	CNTCIDR0	RO	0x0D	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.5 Generic timer control registers

Generic timer control registers provide the programming interface to the control frame of the generic timer.

Table 22: Generic timer control register summary

Offset	Name	Type	Reset	Width	Description
0x000	CNTFRQ (Counter Frequency Register)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x004	CNTNSAR (Counter Non-secure Access Register)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x008	CNTTIDR (Counter Timer ID Register)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x00C - 0x03F	Reserved	**RAZ/WI**	-	-	-
0x040	CNTACR (Counter Access Control Register)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x044 - 0xFCC	Reserved	**RAZ/WI**	-	-	-
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4

Offset	Name	Type	Reset	Width	Description
0xFD4 - 0xFDC	Reserved	RAZ/WI	-	-	-
0xFE0	CNTPIDR0	RO	0xA0	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	0xB0	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	0x0B	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3
0xFF0	CNTCIDR0	RO	0x0D	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.6 Generic timer registers

Generic timer registers provide the programming interface to the generic timer.

Table 23: Generic timer register summary

Offset	Name	Type	Reset	Width	Description
0x000	CNTPCT[31:0] (Physical Count Register low)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x004	CNTPCT[63:32] (Physical Count Register high)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x008	CNTVCT[31:0] (Virtual Count Register low)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*

Offset	Name	Type	Reset	Width	Description
0x00C	CNTVCT[63:32] (Virtual Count Register high)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x010	CNTFRQ (Counter Frequency Register)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x014	Reserved	**RAZ/WI**	-	-	-
0x018	CNTVOFF[31:0] (Virtual Offset Register low)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x01C	CNTVOFF[63:32] (Virtual Offset Register High)	RO	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x020	CNTP_CVAL[31:0] (CompareValue Register low)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x024	CNTP_CVAL[63:32] (CompareValue Register high)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x028	CNTP_TVAL (TimerValue Register)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x02C	CNTP_CTL (Physical Timer Control Register)	RW	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*	32-bit	See *Arm® Architecture Reference Manual Armv7-A and Armv7-R edition*
0x030 - 0xFCC	Reserved	**RAZ/WI**	-	-	-
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	**RAZ/WI**	-	-	-
0xFE0	CNTPIDR0	RO	0xA2	32-bit	Peripheral ID 0

Offset	Name	Type	Reset	Width	Description
0xFE4	CNTPIDR1	RO	0xB0	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	0x0B	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3
0xFF0	CNTCIDR0	RO	0xF0	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.7 Generic watchdog control registers

Generic watchdog control registers provide the programming interface to the control frame of the generic watchdog.

Table 24: Generic watchdog control register summary

Offset	Name	Type	Reset	Width	Description
0x000	WCS (Watchdog Control and Status)	RW	See *Arm® Server Base System Architecture 6.0 Platform Design Document*	32-bit	See *Arm® Server Base System Architecture 6.0 Platform Design Document*
0x004	Reserved	**RAZ/WI**	-	-	-
0x008	WOR (Watchdog Offset)	RW	0x0000_0000	32-bit	Register is full 32-bit word write ONLY
0x00C	Reserved	**RAZ/WI**	-	-	-
0x010	WCV[31:0] (Watchdog compare value low)	RW	0x0000_0000	32-bit	Register is full 32-bit word write ONLY
0x014	WCV[63:32] (Watchdog compare value high)	RW	0x0000_0000	32-bit	Register is full 32-bit word write ONLY
0x018 - 0xFCB	Reserved	**RAZ/WI**	-	-	-

Offset	Name	Type	Reset	Width	Description
0xFCC	Control Frame IIDR	RO	0x0000_143B	32-bit	See Control Frame IIDR
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	RAZ/WI	-	-	-
0xFE0	CNTPIDR0	RO	0xB1	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	0xB0	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	0x2B	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3
0xFF0	CNTCIDR0	RO	0x0D	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.7.1 Control Frame IIDR

Control Frame IIDR contains information to identify the implementer of the watchdog component.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic watchdog control registers

Address offset

0x0000_0FCC

Bit descriptions

Table 25: IIDR bit descriptions

Bits	Types	Reset	Name	Description
31:24	RO	0x00	PRODUCT_ID	Product identification
23:20	RO	0x0	RESERVED	Reserved
19:16	RO	0x0	VARIANT	Architecture version, v0
15:12	RO	0x1	REVISION	Revision Number, r1
11:0	RO	0x43B	IMPLEMENTER	JEP106 Code of company implementing the watchdog

A.2.1.8 Generic watchdog refresh registers

Generic watchdog refresh registers provide the programming interface to the refresh frame of the generic watchdog.

Table 26: Generic watchdog refresh register summary

Offset	Name	Type	Reset	Width	Description
0x000	Watchdog refresh	RW	See *Arm® Server Base System Architecture 6.0 Platform Design Document*	32-bit	See *Arm® Server Base System Architecture 6.0 Platform Design Document*
0x004 - 0xFCB	Reserved	**RAZ/WI**	-	-	-
0xFCC	Refresh Frame IIDR	RO	0x0000_143B	32-bit	See Refresh Frame IIDR
0xFD0	CNTPIDR4	RO	0x04	32-bit	Peripheral ID 4
0xFD4 - 0xFDC	Reserved	**RAZ/WI**	-	-	-
0xFE0	CNTPIDR0	RO	0xB0	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	0xB0	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	0x2B	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	0x00	32-bit	Peripheral ID 3

Offset	Name	Type	Reset	Width	Description
0xFF0	CNTCIDR0	RO	0x0D	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	0xF0	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	0x05	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	0xB1	32-bit	Component ID 3

A.2.1.8.1 Refresh Frame IIDR

Refresh Frame IIDR contains information to identify the implementer of the watchdog component.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic watchdog refresh registers

Address offset

0x0000_0FCC

Bit descriptions

Table 27: IIDR bit descriptions

Bits	Types	Reset	Name	Description
31:24	RO	0x00	PRODUCT_ID	Product identification
23:20	RO	0x0	RESERVED	Reserved
19:16	RO	0x0	VARIANT	Architecture version, v0
15:12	RO	0x1	REVISION	Revision Number, r1
11:0	RO	0x43B	IMPLEMENTER	JEP106 Code of company implementing the watchdog

A.2.2 Example integration layer registers

The example integration layer registers describe information for peripherals in the example integration layer.

Table 28: Integration layer register summary

Name	Type	Reset	Width	Description
I2C	RO, WO	-	-	Inter-Integrated Circuit controller
I2S	RO, WO, RW, **RAZ/WI**	-	32-bit	Inter-IC Sound Bus Controller
CLCD Interface registers	RW	-	-	CLCD Interface

A.2.2.1 I2C, Inter-Integrated Circuit controller

A serial bus module is provided to enable the system to use an APB interface to communicate with the I2C interface.

Configurations

This register is available in all configurations.

Attributes

Width

-

Functional group

Example integration layer registers

Bit descriptions

Table 29: I2C controller bit descriptions

Offset	Name	Type	Reset	Width	Description
0x000	SB_CONTROL	RO	-	-	Read serial control bits: • Bit [0] is SCL • Bit [1] is SDA

Offset	Name	Type	Reset	Width	Description
0x000	SB_CONTROLS	WO	-	-	Set serial control bits: • Bit [0] is SCL • Bit [1] is SDA
0x004	SB_CONTROLC	WO	-	-	Clear serial control bits: • Bit [0] is SCL • Bit [1] is SDA

A.2.2.2 I2S, Inter-IC Sound Bus Controller

An I2S controller module is provided to enable the system to use an APB interface to communicate with the I2S interface.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Example integration layer registers

Bit descriptions

Table 30: I2S controller bit descriptions

Offset	Name	Type	Reset	Width	Description
0x000	CONTROL	RW	0x0000_2200	32-bit	I2S control register
0x004	STATUS	RO	0x0000_0015	32-bit	I2S status register
0x008	ERROR	RW	0x0000_0000	32-bit	I2S error status register
0x00C	DIVIDE	RW	-	32-bit	I2S divide ratio register
0x010	TXBUF	WO	-	32-bit	Transmit buffer FIFO data register

Offset	Name	Type	Reset	Width	Description
0x014	RXBUF	RO	-	32-bit	Receive buffer FIFO data register
0x300	ITCR	RW	-	32-bit	Integration test control register
0x304	ITIP1	RO	-	32-bit	Integration test input register
0x308	ITOP1	RW	-	32-bit	Integration test output register
0xFD0	CNTPIDR4	RO	-	32-bit	Peripheral ID 4
0xFD4	-	**RAZ/WI**	-	-	Reserved
0xFD8	-	**RAZ/WI**	-	-	Reserved
0xFDC	-	**RAZ/WI**	-	-	Reserved
0xFD4	-	**RAZ/WI**	-	-	Reserved
0xFE0	CNTPIDR0	RO	-	32-bit	Peripheral ID 0
0xFE4	CNTPIDR1	RO	-	32-bit	Peripheral ID 1
0xFE8	CNTPIDR2	RO	-	32-bit	Peripheral ID 2
0xFEC	CNTPIDR3	RO	-	32-bit	Peripheral ID 3
0xFF0	CNTCIDR0	RO	-	32-bit	Component ID 0
0xFF4	CNTCIDR1	RO	-	32-bit	Component ID 1
0xFF8	CNTCIDR2	RO	-	32-bit	Component ID 2
0xFFC	CNTCIDR3	RO	-	32-bit	Component ID 3

A.2.2.2.1 I2S control register

The I2S control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 31: I2S Control register bit descriptions

Bits	Types	Reset	Name	Description
31:18	**RAZ/WI**	0x00	RESERVED	Reserved
17	RW	0x0	Audio Codec Reset Control	Audio Codec Reset Control (output pin)
16	RW	0x0	FIFO reset	Fifo reset
15	**RAZ/WI**	0x0	RESERVED	Reserved
14:12	RW	0x2	RX Buffer IRQ Water Level	Default 2 (IRQ trigger when less than 2 word space available)
11	**RAZ/WI**	0x0	RESERVED	Reserved
10:8	RW	0x2	TX Buffer IRQ Water Level	Default 2 (IRQ trigger when more than 2 word space available)
7:4	**RAZ/WI**	0x0	RESERVED	Reserved
3	RW	0x0	RX Interrupt Enable	RX Interrupt Enable
2	RW	0x0	RX Enable	RX Enable
1	RW	0x0	TX Interrupt Enable	TX Interrupt Enable
0	RW	0x0	TX Enable	TX Enable

A.2.2.2.2 I2S status register

The I2S status register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 32: I2S Status register bit descriptions

Bits	Types	Reset	Name	Description
31:6	**RAZ/WI**	0x00	RESERVED	Reserved
5	RO	0x0	RX Buffer Full	RX Buffer Full
4	RO	0x1	RX Buffer Empty	RX Buffer Empty
3	RO	0x0	TX Buffer Full	TX Buffer Full
2	RO	0x1	TX Buffer Empty	TX Buffer Empty
1	RO	0x0	RX Buffer Alert	RX Buffer Alert (Depends on Water level)
0	RO	0x1	TX Buffer Alert	TX Buffer Alert (Depends on Water level)

A.2.2.2.3 I2S error status register

The I2S error status register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 33: I2S Error Status register bit descriptions

Bits	Types	Reset	Name	Description
31:2	**RAZ/WI**	0x00	RESERVED	Reserved
1	RW	0x0	RX Overrun	RX overrun (Write 1 to clear)
0	RW	0x0	TX Overrun	TX overrun/underrun (Write 1 to clear)

A.2.2.2.4 I2S divide ratio register

The I2S divide ratio register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 34: I2S Divide Ratio register bit descriptions

Bits	Types	Reset	Name	Description
31:10	**RAZ/WI**	0x00	RESERVED	Reserved
9:0	RW	0x20	LRDIV	Divide ratio register for Left and Right clock.

A.2.2.2.5 Transmit buffer FIFO data register

The transmit buffer FIFO data register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 35: I2S Transmit Buffer FIFO Data register bit descriptions

Bits	Types	Reset	Name	Description
31:16	WO	–	Left channel	Write to TX buffer left channel
15:0	WO	–	Right channel	Write to TX buffer right channel

A.2.2.2.6 Receive buffer FIFO data register

The receive buffer FIFO data register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 36: I2S Receive Buffer FIFO Data register bit descriptions

Bits	Types	Reset	Name	Description
31:16	RO	0x00	Left channel	Read from RX buffer left channel
15:0	RO	0x00	Right channel	Read from RX buffer right channel

A.2.2.2.7 Integration test control register

The Integration test control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 37: I2S Integration Test Control register bit descriptions

Bits	Types	Reset	Name	Description
31:1	**RAZ/WI**	0x00	RESERVED	Reserved
0	RW	0x0	ITCR	Integration Test Control Register

A.2.2.2.8 Integration test input register

The integration test input register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 38: I2S Integration Test Input register bit descriptions

Bits	Types	Reset	Name	Description
31:1	RAZ/WI	0x00	RESERVED	Reserved
0	RW	0x0	SDIN	Serial Data Line Input

A.2.2.2.9 Integration test output register

The integration test output register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

I2S

Bit descriptions

Table 39: I2S Integration Test Output register bit descriptions

Bits	Types	Reset	Name	Description
31:4	RAZ/WI	0x00	RESERVED	Reserved
3	RW	0x0	IRQOUT	Interrupt Output
2	RW	0x0	LRCK	Left Right Clock

Bits	Types	Reset	Name	Description
1	RW	0x0	SCLK	Serial Clock
0	RW	0x0	SDOUT	Serial Data Line Output

A.2.2.3 CLCD Interface registers

The Color LCD (CLCD) interface is a custom peripheral that provides an interface to a STMicroelectronics STMPE811QTR Port Expander with Advanced Touch Screen Controller on the Keil® MCBSTM32C display board. The Keil display board contains an AM240320LG display panel and uses a Himax HX8347-D LCD controller.

Configurations

This register is available in all configurations.

Attributes

Width

-

Functional group

Example integration layer registers

Bit descriptions

Table 40: CLCD interface

Offset	Type	Name	Description
0x000	RW	Write Command	A write to this address causes a write to the LCD commands register. A read from this address causes a read from the LCD busy register.
0x004	RW	Write data RAM, Read data RAM	A write to this address causes a write to the LCD data register. A read to this address causes a read from the LCD data register.
0x008	RW	Interrupt	Bit 0 indicates Access Complete. Write 0 to Bit 0 to clear. This bit is set if read data is valid. Bits [31:1] can be ignored.

A.3 Interrupt map

Corstone-500 uses hardware and software interrupts.

The following types of interrupts are supported:

Software Generated Interrupts (SGIs)

This interrupt is generated explicitly by software by writing to a dedicated distributor register, the Software Generated Interrupt Register.

Private Peripheral Interrupts (PPIs)

This interrupt is generated by a peripheral that is private to an individual core.

Shared Peripheral Interrupts (SPIs)

This interrupt is generated by a peripheral that the Interrupt Controller can route to more than one core.

A.3.1 Subsystem interrupt map

The following table describes the interrupts in the subsystem.

Table 41: Subsystem interrupt map

Interrupt ID	Description	Type	Trigger	Notes
0-15	Software generated	SGI	-	-
16-26	Reserved	PPI	-	-
27	Global timer	PPI	Edge	-
28	Legacy nFIQ	PPI	Level	-
29	Private timer	PPI	Edge	-
30	Watchdog timers	PPI	Edge	-
31	Legacy nIRQ	PPI	Level	-
32	System Generic Watchdog	SPI	Level	-
33	Reserved	SPI	-	-
34	System Generic Timer - physical	SPI	Level	-
35-37	Reserved	SPI	-	-

Interrupt ID	Description	Type	Trigger	Notes
38	Real-Time Clock	SPI	Level	-
39	True Random Number Generator	SPI	Level	-
40	UART0 combined	SPI	Level	-
41	UART1 combined	SPI	Level	-
42	Debug Interrupt 0	SPI	Level	-
43	Debug Interrupt 1	SPI	Level	-
44	L2C-310 combined interrupt	SPI	Level	-
45	CPU0 debug comm TX	SPI	Level	-
46	CPU0 debug comm RX	SPI	Level	-
47	CPU0 PMU counter overflow	SPI	Level	-
48	CPU0 CTI trigger	SPI	Level	-
49	CPU1 debug comm TX	SPI	Level	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
50	CPU1 debug comm RX	SPI	Level	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
51	CPU1 PMU counter overflow	SPI	Level	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
52	CPU1 CTI trigger	SPI	Level	Implemented if using 2 or more Cortex-A5 cores. Otherwise reserved.
53	CPU2 debug comm TX	SPI	Level	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.
54	CPU2 debug comm RX	SPI	Level	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.

Interrupt ID	Description	Type	Trigger	Notes
55	CPU2 PMU counter overflow	SPI	Level	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.
56	CPU2 CTI trigger	SPI	Level	Implemented if using 3 or more Cortex-A5 cores. Otherwise reserved.
57	CPU3 debug comm TX	SPI	Level	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.
58	CPU3 debug comm RX	SPI	Level	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.
59	CPU3 PMU counter overflow	SPI	Level	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.
60	CPU3 CTI trigger	SPI	Level	Implemented if using 4 or more Cortex-A5 cores. Otherwise reserved.
61-63	Reserved	SPI	-	-

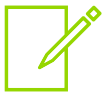
A.3.2 Example integration layer interrupt map

The following table describes the interrupts in the integration layer.

Table 42: Integration layer interrupt map

Interrupt ID	Description	Type	Trigger	Notes
64	UART2 combined	SPI	Level	-
65	UART3 combined	SPI	Level	-
66	UART4 combined	SPI	Level	-
67	UART5 combined	SPI	Level	-
68	I2S0	SPI	Level	-

Interrupt ID	Description	Type	Trigger	Notes
69	I2S1	SPI	Level	-
70	SPI0 combined	SPI	Level	-
71	SPI1 combined	SPI	Level	-
72	SPI2 combined	SPI	Level	-
73	CLCD	SPI	Level	-
74	USB	SPI	Level	-
75	Ethernet	SPI	Level	-
76	GPIO0 combined	SPI	Level	-
77	GPIO1 combined	SPI	Level	-
78	GPIO2 combined	SPI	Level	-
79	GPIO3 combined	SPI	Level	-
80	GPIO4 combined	SPI	Level	-
81	GPIO5 combined	SPI	Level	-
82	GPIO6 combined	SPI	Level	-
83	GPIO7 combined	SPI	Level	-
84	FPGA_generic	SPI	Level	-
85-255	Reserved	SPI	-	-


Note

In the FPGA, interrupt 84 (FPGA_generic) is connected to an inverted copy of the CLCD_TINT signal from the MPS3 board. In the MPS3 TRM, this signal is called LCD_TSINT

Appendix B Revisions

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

Table 43: Issue 0000_01

Change	Location
First Non-Confidential Early Access Release for r0p0	-