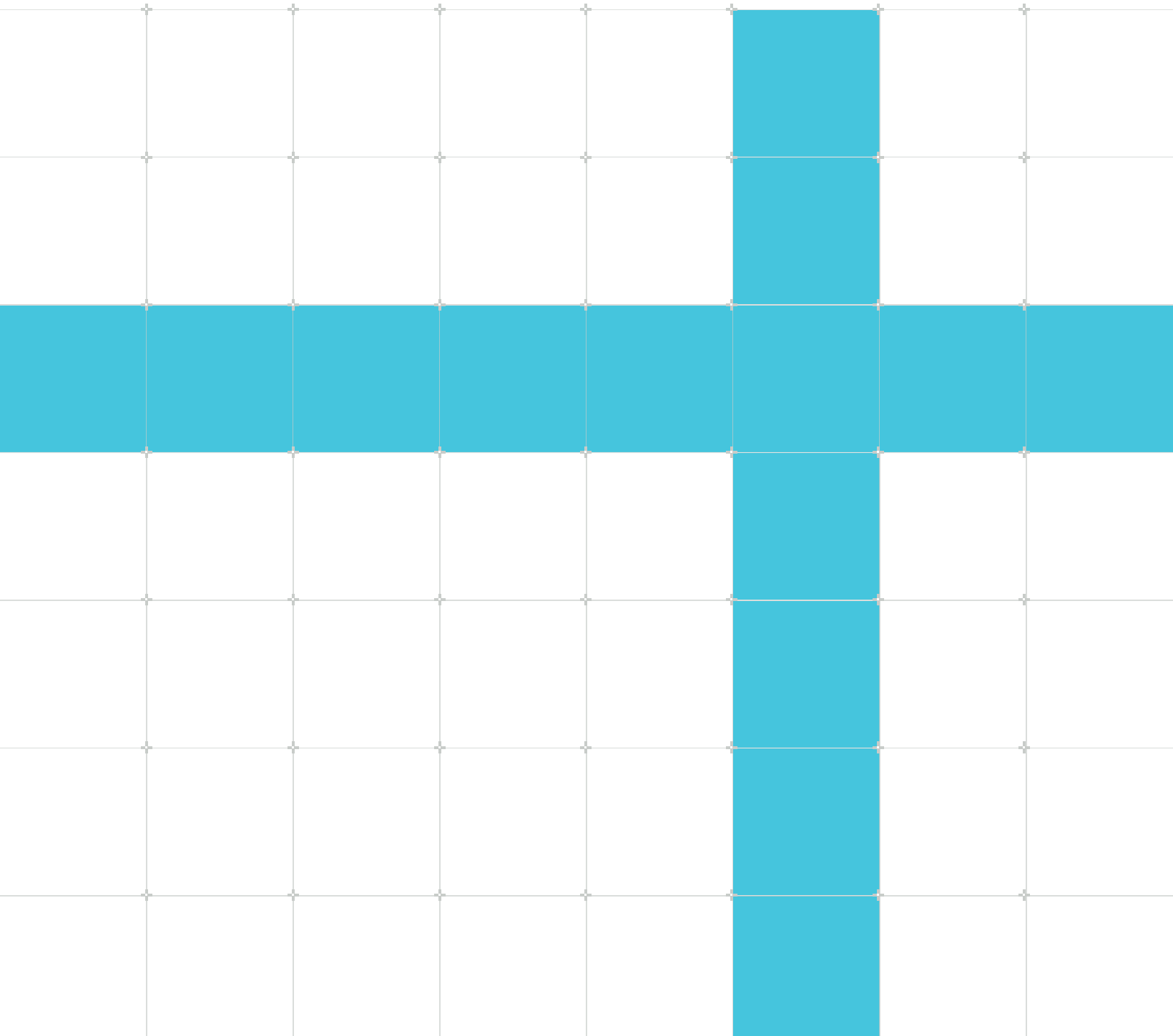
Arm® Total Compute 2022 Reference Design

Software Developer Guide

Issue 01
108028_0000_01_en



Arm® Total Compute 2022 Reference Design

Software Developer Guide

Release information

Document history

Issue	Date	Confidentiality	Change
0000-01	8 September 2023	Non-Confidential	First release

Proprietary Notice

and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

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

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

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

Unrestricted Access is an Arm internal classification.

Product Status

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

Feedback

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

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

Inclusive language commitment

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

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

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

Contents

1. Introduction

1.1 Conventions

The following subsections describe conventions used in Arm documents.

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.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	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	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.
Caution	Recommendations. Not following these recommendations might lead to system failure or damage.
Warning	Requirements for the system. Not following these requirements might result in system failure or damage.
Danger	Requirements for the system. Not following these requirements will result in system failure or damage.
Note	An important piece of information that needs your attention.

Convention	Use
 Tip	A useful tip that might make it easier, better or faster to perform a task.
 Remember	A reminder of something important that relates to the information you are reading.

1.2 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® CoreLink™ CI-700 Coherent Interconnect Technical Reference Manual	101569	Non-Confidential
Arm® CoreLink™ GIC-700 Generic Interrupt Controller Technical Reference Manual	101516	Non-Confidential
Arm® CoreLink™ MMU-700 System Memory Management Unit Technical Reference Manual	101542	Non-Confidential
Arm® CoreLink™ NI-700 Network-on-Chip Interconnect Technical Reference Manual	101566	Non-Confidential
Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual	DDI 0475	Non-Confidential
Arm® CoreLink™ NIC-450 Network Interconnect Technical Overview	100459	Non-Confidential
Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual	101150	Non-Confidential
Arm® CoreLink™ TZC-400 TrustZone® Address Space Controller Technical Reference Manual	100325	Non-Confidential
Arm® CoreSight™ Components Technical Reference Manual	DDI 0314	Non-Confidential
Arm® CoreSight™ STM-500 System Trace Macrocell Technical Reference Manual	DDI 0528	Non-Confidential
Arm® CoreSight™ System-on-Chip SoC-600 Technical Reference Manual	100806	Non-Confidential
Arm® CoreSight™ Trace Memory Controller Technical Reference Manual	DDI 0461	Non-Confidential
Arm® Cortex®-A520 Core Cryptographic Extension Technical Reference Manual	102519	Non-Confidential
Arm® Cortex®-A520 Core Technical Reference Manual	102517	Non-Confidential
Arm® Cortex®-A720 Core Cryptographic Extension Technical Reference Manual	102532	Non-Confidential
Arm® Cortex®-A720 Core Technical Reference Manual	102530	Non-Confidential
Arm® Cortex®-M System Design Kit Technical Reference Manual	DDI 0479	Non-Confidential
Arm® Cortex®-M3 Processor Technical Reference Manual	100165	Non-Confidential
Arm® Cortex®-X4 Core Core Cryptographic Extension Technical Reference Manual	102486	Non-Confidential
Arm® Cortex®-X4 Core Core Technical Reference Manual	102484	Non-Confidential
Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual	102547	Non-Confidential
Arm® Mali™-G720 GPU Model Integration Guide	102763	Confidential
Arm® Mali™-G720 GPU Technical Reference Manual	102760	Confidential

Arm product resources	Document ID	Confidentiality
Fast Models Fixed Virtual Platforms (FVP) Reference Guide	100966	Non-Confidential
Fast Models Reference Guide	100964	Non-Confidential
PrimeCell UART (PL011) Technical Reference Manual	DDI 0183	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
AMBA® 5 CHI Architecture Specification	IHI 0050E.a	Non-Confidential
AMBA® AXI and ACE Protocol Specification	IHI 0022	Non-Confidential
AMBA® CXS Protocol Specification	IHI 0079B	Non-Confidential
AMBA® Low Power Interface Specification	IHI 0068	Non-Confidential
Arm® Architecture Reference Manual for A-profile architecture	DDI 0487	Non-Confidential
Arm® Base System Architecture 1.0 Platform Design Document	DEN 0094	Non-Confidential
Arm® CoreSight™ Architecture Specification v3.0	IHI 0029	Non-Confidential
Arm® CoreSight™ Base System Architecture 1.0, Arm Platform Design Document	DEN 0068.v	Non-Confidential
Arm® CoreSight™ System Control and Management Interface Platform Design Document	DEN 0056	Non-Confidential
Arm® Debug Interface Architecture Specification ADIv5.0 to ADIv5.2	IHI 0031	Non-Confidential
Arm® Debug Interface Architecture Specification ADIv6.0	IHI 0074	Non-Confidential
Arm® Generic Interrupt Controller Architecture Specification, GIC architecture version 3 and version 4	IHI 0069	Non-Confidential
Arm® Power Control System Architecture, version 2.1	DEN 0050	Non-Confidential
Arm® Power Policy Unit Architecture Specification, version 1.1	DEN 0051E	Non-Confidential
Arm® System Memory Management Unit Architecture Specification, SMMU architecture version 3.0, 3.1 and 3.2	IHI 0070	Non-Confidential
Arm®v7-M Architecture Reference Manual	DDI 0403	Non-Confidential
Trusted Base System Architecture, Client (4th Edition)	DEN 0021D	Non-Confidential

Non-Arm resources	Document ID	Organization
Ray Tracing in Vulkan®	–	Khronos® Group

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at http://www.adobe.com

1.3 Other information

See the Arm website for other relevant information.

- Arm® Developer.
- Arm® Documentation.
- Technical Support.
- Arm® Glossary.

2. Overview of Total Compute 2022 Reference Design

Arm® Total Compute Reference Design describes and models the design choices for recommended configuration of typical Arm-based subsystems. RD-TC22 incorporates 2022-generation products from Arm, and other system IP, along with an OSS stack, targeted at the client markets such as smartphones, laptops, and set-top boxes.

The RD-TC22 Fixed Virtual Platform (FVP) is a functional model intended for software development on the recommended subsystem configurations. For more information on supported reference configuration, see RD-TC22 Fixed Virtual Platform.

About Arm Development tools

Arm tools and models are designed as Arm® Success Kits for partner access, supporting hardware SoC development and software development. Success Kits are available as Hardware Success Kits for SoC development and Software Success Kits for software development.

To understand what is included in Success Kits and their relevant use cases, see the Data Sheet, or visit Arm Success Kits for more information.

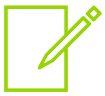
2.1 Deliverables

The following deliverables are provided to enable software development.

Total Compute 2022 Reference Design Software Developer Guide

A Software Developer Guide providing a high-level overview of RD-TC22, including the architecture from which the RD-TC22 design has been derived, associated software stack, and the FVP.

The following RD-TC22 confidential documents are available for developers building best in class mobile SoC-based on Arm architecture. Contact Arm to obtain access to full set of RD-TC22 documentation.



The documentation set includes:

- *Arm® Total Compute 2022 Reference Design Getting Started Guide*
- *Arm® Total Compute 2022 Reference Design System Design*
- *Arm® Total Compute 2022 Reference Design Technical Overview*
- *Arm® Total Compute 2022 Reference Design Performance Analysis Report*
- *Arm® Total Compute 2022 Reference Design Power Analysis Report*

- *Arm® Total Compute 2022 Reference Design FPGA Solution Performance Analysis Report*

Fixed Virtual Platform

A *Fixed Virtual Platform* (FVP) providing a software model of the RD-TC22 reference mobile configuration design. It models the programmers' view of the design that allows to execute software without an actual hardware platform. The execution speeds that are available makes it possible to run a full OS, such as Linux and Android™, on the model. The Total Compute software stack can directly execute on the FVP. For more information on the FVP, see Total Compute on the Arm Developer website.

Total Compute Reference Software Stack

A software stack that covers all the necessary software components for a client platform, from firmware at the bottom, up to the OS at the top. It also includes all the Secure software components that run in the Secure world of the design. The software stack provides a starting point to modify, extend, and develop the software stack for a *System on Chip* (SoC) similar to RD-TC22. The FVP is used with the Total Compute reference software stack. For instructions on software, and how to set up and run the FVP, see Total Compute on the Arm Developer website.

2.2 Features of RD-TC22

RD-TC22 supports the Arm®v9.2-A architecture that extends the architecture defined in Arm®v8-A architectures up to Arm®v8.7-A.

It also provides key features built around the following 2022-generation IP:

- A single processor cluster, up to a maximum of 14 cores in a combination of the following cores with DSU-120:
  - Arm® Cortex®-X4 Core
  - Arm® Cortex®-A720 Core
  - Arm® Cortex®-A520 Core
- Arm® Mali™-G720 GPU to support ray tracing. For more information on ray tracing, see *Ray Tracing in Vulkan®*.

For a full list of Arm IP used in the FVP, see RD-TC22 Fixed Virtual Platform.

For an overview of block-level involvement and features of all the IP, see System architecture.

2.3 Compliance

The Total Compute 2022 Reference Design complies with, or includes components that comply with, the following specifications:

- *Arm® Architecture Reference Manual for A-profile architecture*
- *Arm® Generic Interrupt Controller Architecture Specification, GIC architecture version 3 and version 4*
- Arm® AMBA specifications
- *Arm® CoreSight™ Base System Architecture 1.0, Arm Platform Design Document*
- *Arm® Power Policy Unit Architecture Specification, version 1.1*
- *Arm® System Memory Management Unit Architecture Specification, SMMU architecture version 3.0, 3.1 and 3.2*
- *Trusted Base System Architecture, Client (4th Edition)*


Remember

All relevant specifications and their document IDs are listed in the *Useful resources* section of the Introduction chapter.

Confidential documents are only available to licensees through the product package, so links to online versions are not available in this document.

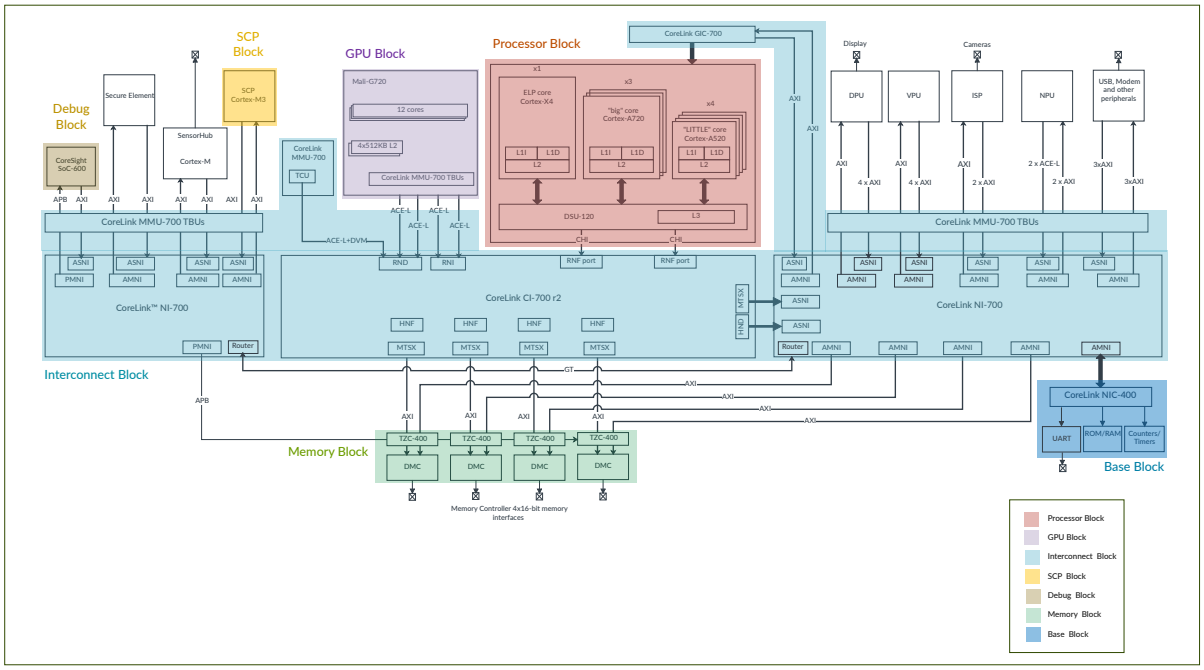
3. System architecture

The reference subsystems of RD-TC22 are partitioned into functional blocks that are a combination of major IP and the supporting logic around it.

Some features of the design incorporate functionality from multiple blocks. The block-based design approach provides flexibility, scalability, and modularity.

The following figure gives a high-level architectural view of the Mobile reference subsystem.

Figure 3-1: System Architecture



3.1 Processor Block

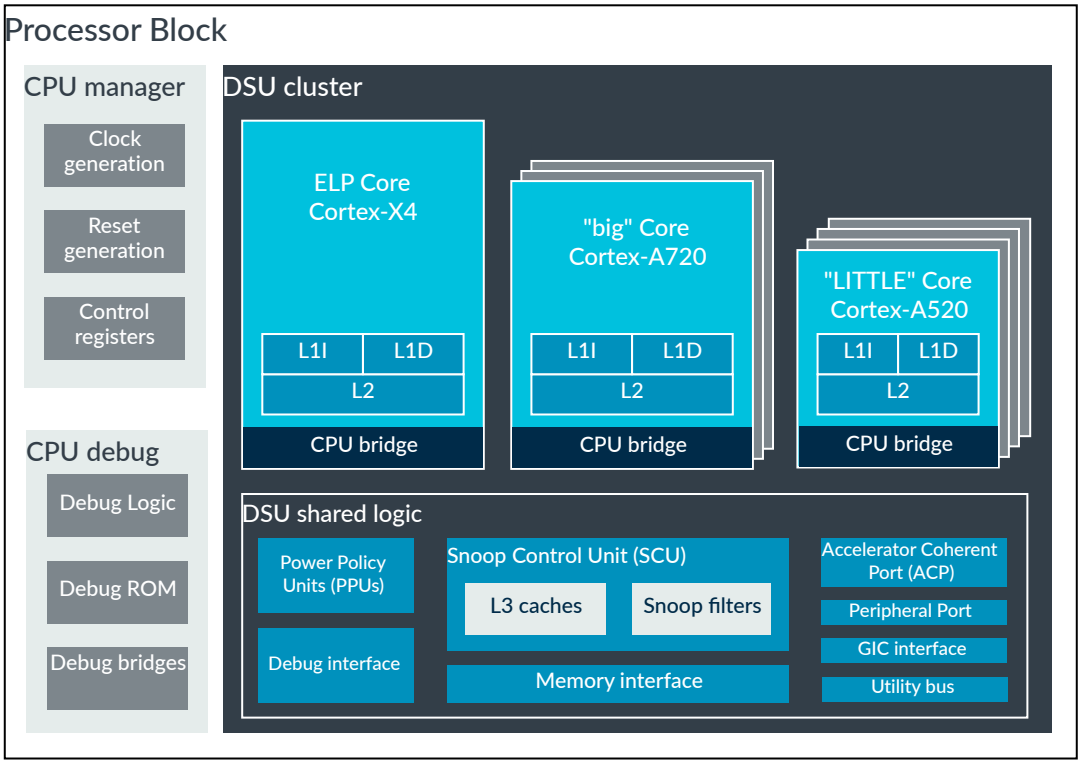
The main functional part of the Processor Block is a core cluster which consists of cores wrapped in the DynamIQ™ Shared Unit (DSU). It also contains other supporting logic, including processor control, processor debug, and domain bridges.

One Processor Block is incorporated inside a RD-TC22 reference subsystem. The type and number of cores are configured differently for different subsystems. The cores in the Processor Block are also referred to as *Application Processors* (APs).

Three tiers of cores can be used in the block for optimizing both performance and power. The DSU provides shared L3 cache and other logic among the cores.

The following figure shows the block diagram of the Processor Block.

Figure 3-2: Block diagram of Processor Block



3.1.1 Cores

Different tiers of Arm® v9.2-A cores are used in the Processor Block to achieve optimal balance between performance and power consumption while running various workloads.

For the premium solution, three tiers of cores are used:

ELP core: Cortex®-X4 Core

Designed for pushing performance beyond the “big” core by adding extra hardware resources compared to the “big” core. It boosts the maximum single-thread performance of the premium tier subsystem.

“big” core: Cortex®-A720 Core

Designed for high performance while maintaining a certain level of power efficiency. It does the heavy lifting and improves the overall performance of the subsystem.

“LITTLE” core: Cortex®-A520 Core

Designed for high-efficiency, low power. It improves the subsystem efficiency while the workload is low.

Cortex®-X4 Core

- *Arm® Cortex®-X4 Core Core Technical Reference Manual*
- *Arm® Cortex®-X4 Core Core Cryptographic Extension Technical Reference Manual*

Cortex®-A720 Core

- *Arm® Cortex®-A720 Core Technical Reference Manual*
- *Arm® Cortex®-A720 Core Cryptographic Extension Technical Reference Manual*

Cortex®-A520 Core

- *Arm® Cortex®-A520 Core Technical Reference Manual*
- *Arm® Cortex®-A520 Core Cryptographic Extension Technical Reference Manual*

3.1.2 DSU cluster

The DynamIQ™ Shared Unit, DSU-120 provides a shared L3 memory system, snoop control and filtering, and other control logic to support a cluster of A-class architecture cores. The cluster is called a DSU cluster.

There are several other DSU parameters for configuring the number of register slices between cores to the DSU and the links within DSU transport. They are configured for facilitating implementation timing closure, and based on feedback from implementation trial.

The L3 cache system and external system interconnect (CI-700) are clocked synchronously in integer multiple ratio to minimize memory access latency.

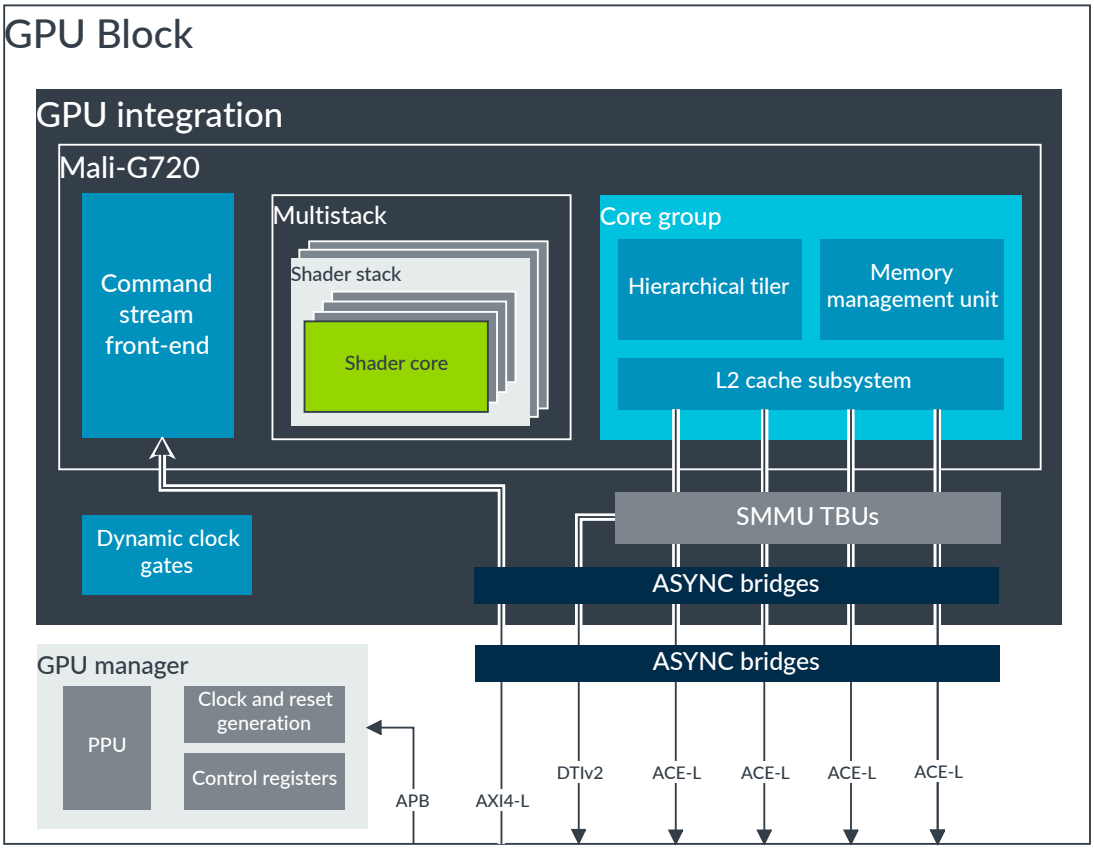
For more information on the DSU and its parameters, see the *Arm® DynamIQ™ Shared Unit-120 Technical Reference Manual*.

3.2 GPU Block

The GPU Block consists of the Mali™-G720 GPU, *Translation Buffer Unit* (TBU) of the *System Memory Management Unit* (SMMU), and other supporting logic.

The following figure shows the block diagram of the GPU Block.

Figure 3-3: Block diagram of GPU Block



As shown in the preceding diagram, the Mali™-G720 GPU is the main component. It supports configurable shader cores for different *Power, Performance, and Area* (PPA) targets. The Mali™-G720 GPU also supports Arm® graphic compression technology, which significantly reduces its bandwidth to main memory, reducing power consumption and improving overall system performance.

For more information on the Mali™-G720 GPU, see the following documents:

- *Arm® Mali™-G720 GPU Technical Reference Manual*
- *Arm® Mali™-G720 GPU Model Integration Guide*

Remember

All relevant document IDs are listed in the *Useful resources* section of the Introduction chapter.

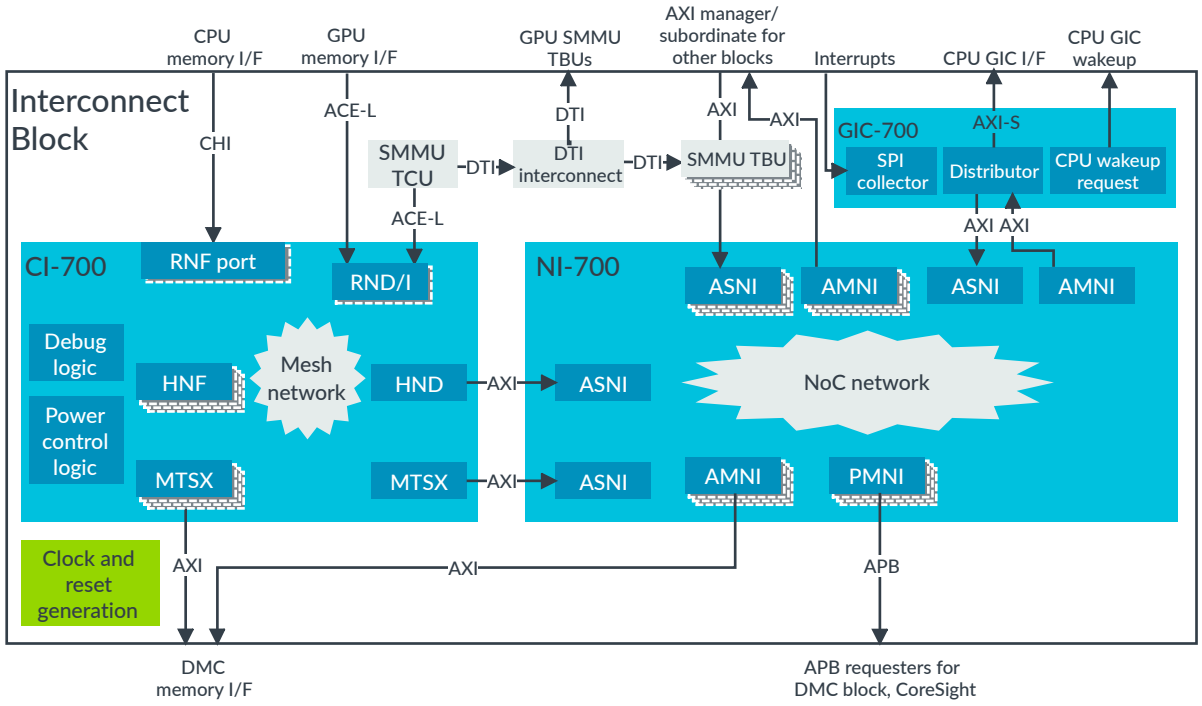
Confidential documents are only available to licensees through the product package, so links to online versions are not available in this document.

3.3 Interconnect Block

The main function of the Interconnect Block is to provide data paths between all other blocks through its coherent and non-coherent interconnect components. It also contains the *Generic Interrupt Controller* (GIC) for application processor interrupt generation, and the *System Memory Management Unit* (SMMU) for address translation.

The following figure shows the block diagram of the Interconnect Block.

Figure 3-4: Block diagram of Interconnect Block



The CI-700 is the coherent interconnect used inside the Interconnect Block. Its HN-F nodes contain *System Level Cache* (SLC) that is fully coherent to the memory inside the *DynamIQ™ Shared Unit* (DSU), and I/O coherent to the memory inside the GPU. This large on-die SLC reduces the latency of communication between the host, GPU, and other blocks, and lowers power consumption by reducing external main memory access.

NI-700 is the non-coherent interconnect used in the Interconnect Block. It is based on a *Network-on-Chip* (NoC) concept which makes it highly flexible and configurable. It is used as the backplane interconnect to connect all the blocks except for the memory interfaces of the DSU, GPU, and SMMU *Translation Control Unit* (TCU). The topology of the NI-700 is designed to match subsystem implementation floorplan requirements.

The CoreLink™ GIC-700 provides interrupt services to the application processor. The GIC Distributor communicates with the GIC Cluster Interface located inside the DSU through an *Advanced eXtensible Interface 4 Stream* (AXI4-stream) link.

The SMMU provides address translation to requesters which do not have a built-in MMU. It uses a distributed structure which separates the *Translation Control Unit* (TCU) and *Translation Buffer Unit* (TBU) for easy integration and implementation. The TCU and TBUs are connected using *Direct Translation Interface* (DTI) links.

- For more information on CoreLink™ CI-700 r2, see the *Arm® CoreLink™ CI-700 Coherent Interconnect Technical Reference Manual*.
- For more information on CoreLink™ NI-700 r2, see the *Arm® CoreLink™ NI-700 Network-on-Chip Interconnect Technical Reference Manual*.
- For more information on CoreLink™ GIC-700, see the *Arm® CoreLink™ GIC-700 Generic Interrupt Controller Technical Reference Manual*.
- For more information on CoreLink™ MMU-700, see the *Arm® CoreLink™ MMU-700 System Memory Management Unit Technical Reference Manual*.

3.4 SCP Block

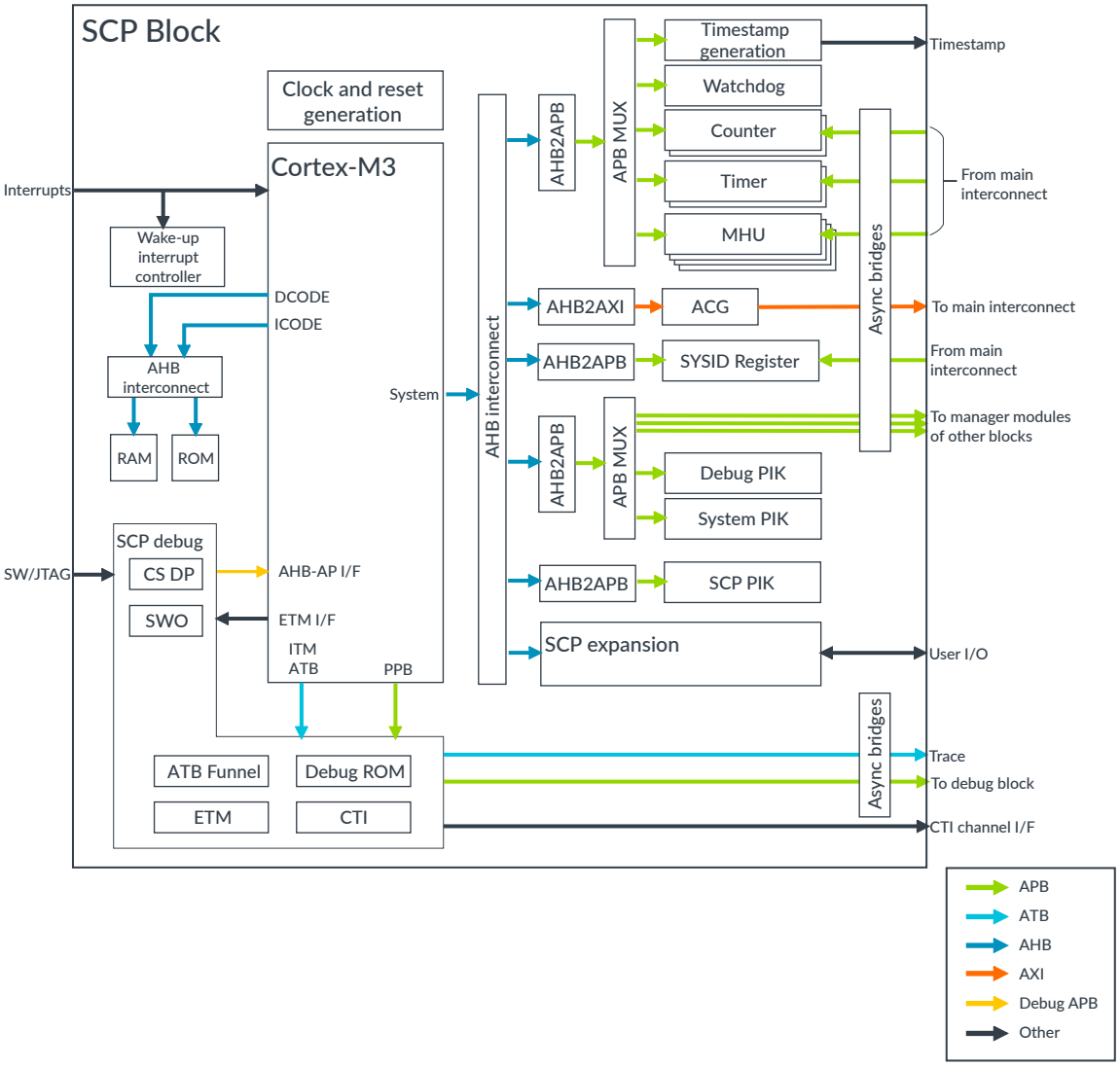
The *System Control Processor* (SCP) Block is an always-on block inside the reference design. It is mainly responsible for low-level system management, including the following listed points.

- Boot and system start-up
- Initial static configuration
- Managing transitions between operating points, that is, an external voltage regulator and clock management to support *Dynamic Voltage and Frequency Scaling* (DVFS)
- Handling hardware wake-up requests from components like timers and other components capable of generating interrupts
- Thermal sensor reading and processing
- Saving and restoring all states in the interconnect when powering down or powering up
- Responsible for managing a consistent matrix of device states across the whole system

Apart from these functional features, the block acts as the root of the chain of trust for *Trusted Base System Architecture Platform Design Document* compliance. An off-chip debug interface, *Joint Test Action Group* (JTAG), is also provided.

The following figure shows the block diagram of the SCP Block.

Figure 3-5: Block diagram of SCP Block



The core of the SCP is a Cortex®-M3 processor. It has RAM, ROM, clock and reset generation, and other necessary peripheral, such as timers and counters, to support its execution.

It controls reset, clock, and power through the *Power Integration Kits* (PIKs) located inside the SCP Block and manager modules inside other blocks.

During the boot-up process, the SCP also initiates system components, for example, the CI-700 and DMC, on the *Application Processor* (AP) side through its connection to the main Interconnect Block.

An expansion module is provided for users adding SoC-specific storage interfaces and other SCP peripherals, such as flash controllers, I2C controllers, and SPI controllers. *Advanced High-performance Bus* (AHB) data bus, interrupts, clock, reset, and power control signals (Q-Channels) are provided to the module.

For more information on Cortex®-M3, see the following documents:

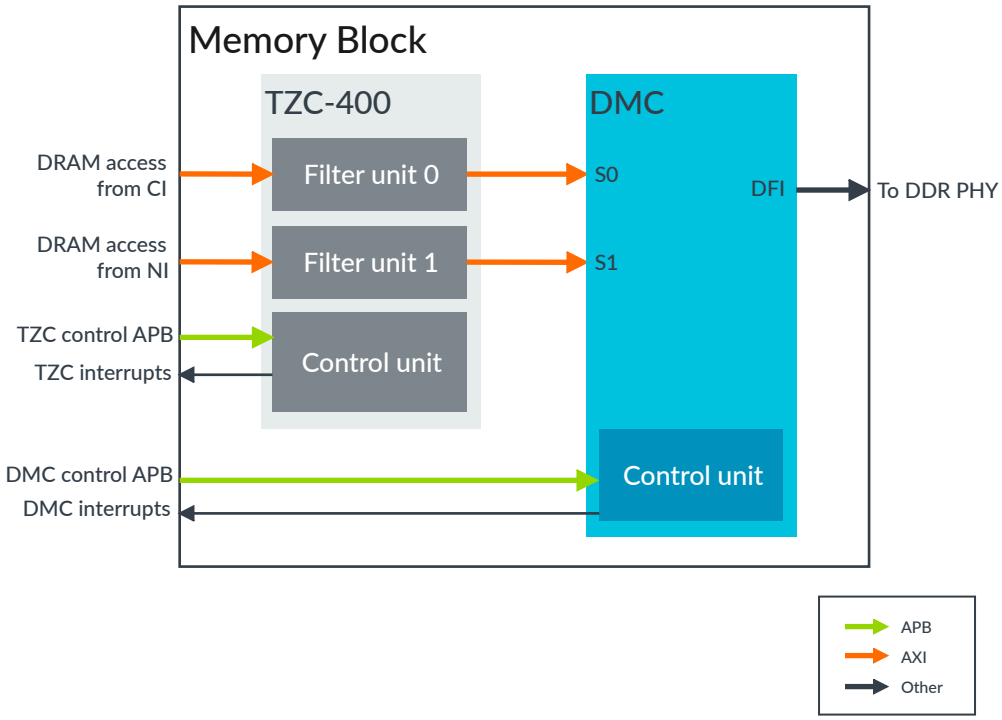
- *Arm® Cortex®-M3 Processor Technical Reference Manual*
- *Arm® Cortex®-M System Design Kit Technical Reference Manual*

3.5 Memory Block

The Memory Block provides DRAM access to the reference subsystems. It contains TZC-400 controllers for TrustZone security checks, and *DRAM Memory Controller* (DMC) for DRAM access.

The following figure shows the block diagram of the Memory Block.

Figure 3-6: Block diagram of Memory Block



The Memory Block contains *Dynamic Memory Controller* (DMC) and Arm® CoreLink™ TZC-400. It wraps these components with the necessary glue-logic to be integrated with the rest of the subsystem.

Each Memory Block provides one X16 DRAM channel, therefore multiple blocks are needed based on the subsystem main memory configuration. Each Memory Block has two subordinate *Advanced eXtensible Interface* (AXI) interfaces accepting memory access traffic from the CI-700 and NI-700 of the Interconnect Block respectively. The main output of the Memory Block is a DFI5.0 interface, to which an appropriate *Double Data Rate* (DDR) *Physical layer* (PHY) is connected to outside of the reference subsystems.

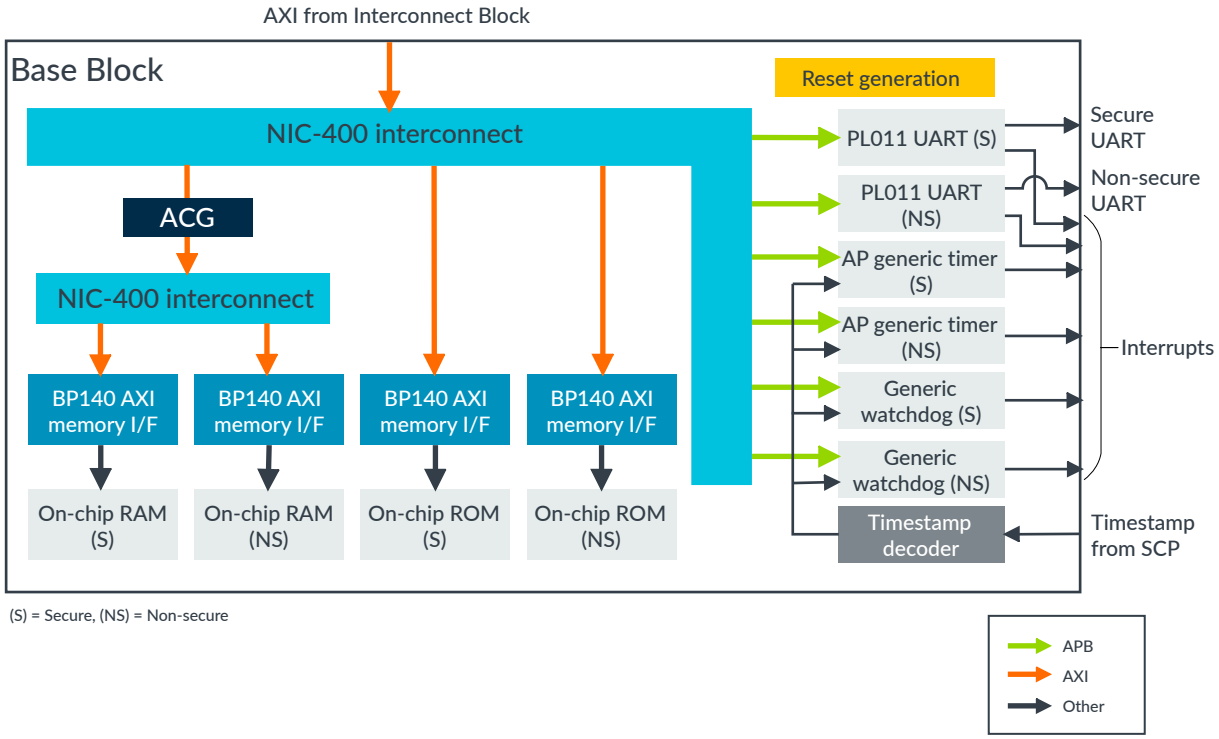
For more information on CoreLink™ TZC-400, see the *Arm® CoreLink™ TZC-400 TrustZone® Address Space Controller Technical Reference Manual*.

3.6 Base Block

The Base Block contains on-chip memory and system peripherals for *Application Processors* (APs).

The following figure shows the block diagram of the Base Block.

Figure 3-7: Block diagram of Base Block



Note

The NIC-400 is part of the CoreLink™ NIC-450 Network Interconnect IP bundle.

As shown in the figure, APs can access memory and peripherals through the *Advanced eXtensible Interface* (AXI) connected to the system main interconnect. For supporting Arm® TrustZone® technology, two sets of memory and peripherals are included in the block to support Secure and Non-secure applications.

The *Access Control Gate* (ACG) shown in the figure supports RAM retention. It stops upstream traffic when RAM is put into the retention state. It can also be programmed to request RAM to exit retention when traffic comes from upstream.

The following table shows a full list of memories and peripherals contained in the block.

Table 3-1: Memories and system peripherals of Base Block

Name	Description
Secure ROM	Contains the code for initializing the boot process. It is accessible only in the Secure mode. For more information on memories and peripherals, see AP memory map
Secure RAM	Scratch RAM used by Secure applications processor software. It is accessible only in the Secure mode. For more information on memories and peripherals, see AP memory map
Non-secure ROM	Contains code required during firmware updates. It is accessible in both Secure and Non-secure modes. For more information on memories and peripherals, see AP memory map
Non-secure RAM	Scratch RAM used by Non-secure applications processor software. It is accessible in both Secure and Non-secure modes. For more information on memories and peripherals, see AP memory map
Secure UART	PrimeCell PL011 *Universal Asynchronous Receiver Transmitter* (UART) It is accessible only in the Secure mode.
Non-secure UART	PrimeCell PL011 UART It is accessible in both Secure and Non-secure modes.
Secure Watchdog	Generic Watchdog Timer for Secure applications
Non-secure Watchdog	Generic Watchdog Timer for Non-secure applications
Secure timer	Generic for Secure applications
Non-secure timer	Generic for Non-secure applications

For more information on the PrimeCell PL011 UART, see the *PrimeCell UART (PL011) Technical Reference Manual*.

For more information on the NIC-400 Interconnect, see the *Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual*.

4. Functional description

Components in RD-TC22, such as clocks, power management, timers, and counters, work across multiple functional blocks.

4.1 Clocks

The RD-TC22 reference subsystems use both external and internal clocks.

4.1.1 Input clocks

Input clocks are clocks provided to the subsystems from the outside. They must be provided with a source that is external to, or embedded in, the subsystem.

The following table lists the input clocks.

Table 4-1: Input clocks

Clock name	PLL lock signal	Description
REFCLK	-	Subsystem main reference clock, which *System Control Processor* (SCP) boots coming out of reset. It must always be running while the system is in use. It is expected to be between 25MHz and 100MHz except in *CSS.SLEEP1* state where it must be externally switched to 32KHz.
SWCLKTCK	-	SWCLKTCK is the clock that drives the combined JTAG and *Serial Wire Debug* (SWD) interface. This interface provides connectivity to an external debugger.
MEMCLK<n>	-	Memory interface clocks associated with the DFI interfaces between *Dynamic Memory Controller* (DMC) and *Double Data Rate* (DDR) *Physical Layer* (PHY). One clock per DFI interface, where *n* is the number of DFI interfaces in a subsystem. Where *n* = 4
EMCLK<n>	-	Clock inputs for external requester expansion I/O ports with configurable *Translation Buffer Unit* (TBU) and *Interrupt Translation Services* (ITS) present on the expansion I/O path. *n* represents the number of I/O ports, which is configurable in the subsystem. There are also corresponding clock Q-Channels associated to each port. They enable dynamic clock gating on the side of external requesters.

4.1.2 Output clocks

Output clocks are clocks provided by subsystems and used by external entities located either on or off the chip contained in the subsystem.

The following table lists the output clocks. For clocks which support dynamic clock gating, both the gated clock and free running clock of the clocks, along with associated clock Q-Channels are provided.

Table 4-2: Output clocks

Clock name	Dynamic clock gating support	Description
EXTCLKOUT EXTCLKOUTFREE	Yes	Output clock for system expansion interfaces
GICCLKOUT GICCLKOUTFREE	Yes	Output clock for synchronizing external edge interrupts input into this subsystem. They should be synchronized to this clock and be one clock cycle wide pulse.
SCPHCLKOUT SCPHCLKOUTFREE	Yes	Output clock for SCP expansion module. See Note (a). It is associated with the SCP expansion the *Advanced High-performance Bus* (AHB) to that module.
ATCLKDBG	No	CoreSight ATB interface clock. It is for external ATB interfaces.
TRACECLK	No	Output clock for Trace Output Port Interface


Note

1. The SCP expansion module is modifiable by partners. Its boundary is treated as part of the boundary of the subsystem. Therefore, SCPHCLKOUT and SCPHCLKOUTFREE are considered as output clocks of the subsystem.

4.2 Power management

Good power management is key to reducing system power consumption while maintaining high performance.

An RD-TC22 reference subsystem contains the following power management features:

- Multiple voltage domains to allow for *Dynamic Voltage and Frequency Scaling* (DVFS) on application processors and the GPU
- Multiple power-gated regions provide comprehensive leakage management
- Multiple power states for different system scenarios
- A *System Control Processor* (SCP) based on a Cortex®-M3 processor controls power, clock, reset, and the static configuration of the system
- *Power Policy Units* (PPUs) are used to manage power states of each voltage and power domain under the control of the SCP

4.2.1 Voltage domains

A voltage domain is defined as a collection of design elements that share a single main voltage supply. The voltage supply to the domain may be scaled or switched off for power or performance reasons.

A Total Compute 2022 Reference Design (RD-TC22) reference subsystem contains the following voltage domains:

VCPU0
The first voltage domain in the processor cluster for “LITTLE” cores. Supports DVFS.

VCPU1
The second voltage domain in the processor cluster for ELP and “big” cores. Supports DVFS.

VGPU
The voltage domain for the GPU. Supports DVFS.

VSYS
The voltage domain for the rest of the subsystem. Does not support DVFS.

4.2.2 Power domains

A power domain is a collection of design elements within a voltage domain that share a common power strategy. A voltage domain can contain one or more power domains.

A power-gated domain is a power domain, the power of which can be removed by on-chip power switches.

A Total Compute 2022 Reference Design (RD-TC22) reference subsystem contains the following top-level power domains:

- DBGTOP is the power domain for Debug-related logic. This domain is a power-gated domain.
- AONTOP is the power domain for the always-ON part of the logic. This domain is a separate power domain but not a power-gated domain.
- CLUS0TOP is the power domain for the processor cluster. This is a power-gated domain.
- GPUTOP is the power domain for the GPU top. This is a power-gated domain.
- SYSTOP is the power domain for the rest of the subsystem. This is a power-gated domain.

Within each top-level power domain, there are additional lower-level power domains defined by the respective blocks.

4.3 Timers and counters

Time values and time events are important resources inside a computer system. They are used to schedule temporal activities and record time events. As defined in the *Arm® Architecture Reference Manual for A-profile architecture*, counters generate time values based on clock signals, and timers generate time events based on time values.

REFCLK time domain
The view of time observed by the *Application Processors* (APs) and *System Control Processor* (SCP).

APs operate in a time domain referred to as REFCLK time. This time domain is based on the main reference clock, REFCLK. This time domain is also visible to the SCP.

A Generic Counter, referred to as the REFCLK Counter, generates time values for the REFCLK time domain. The component meets the requirements of the memory-mapped counter module that is described in the *Arm® Architecture Reference Manual for A-profile architecture*. It is in the VSYS.AONTOP power domain.

This time domain can be halted during debug.

The REFCLK time domain contains several timers:

- All APs in the subsystem implement the Arm Generic Timer that is defined by the *Arm® Architecture Reference Manual for A-profile architecture*. The interrupts from these timers are mapped to *Private Peripheral Interrupts* (PPIs) through the *Generic Interrupt Controller* (GIC). You can access the Generic Timers through a low-latency CP15 register interface.
- SCP_REFCLK generic timer, for use by the SCP. See SCP_REFCLK generic timer.
- Two additional REFCLK generic timers, one Secure and one Non-secure, for use by APs.
- REFCLK drives the GPU and VPU Global Timestamp Counter inputs, and they belong to the REFCLK time domain.

When all components that can observe the REFCLK time domain are in OFF or MEM_RET power modes, the SCP can disable the REFCLK clock. This makes the REFCLK time domain unavailable.

After restoring REFCLK, the SCP firmware must ensure consistent time values are available in the REFCLK Counter. This must be done before putting the subsystem into a state in which a component can observe REFCLK time, which includes putting such a component into the ON power mode.

REFCLK counter

The REFCLK counter is an implementation of the memory-mapped counter module that is defined in the *Arm® Architecture Reference Manual for A-profile architecture*. The counter is visible in both the AP and SCP memory maps, where they are labeled REFCLK CNTControl and REFCLK CNTRead. The counter is implemented in the VSYS.AONTOP power domain. Access to the CNTControl frame is Secure.

This counter can be halted during debug using the cross trigger network.

4.3.1 Timers

There are different timers used in Total Compute 2022 Reference Design (RD-TC22) reference subsystems. They are grouped into two types, generic timers and watchdog timers.

SCP_REFCLK generic timer

The SCP Block has a memory-mapped timer that is only accessible to the SCP. The timer is in the VSYS.AONTOP power domain. The timer conforms to the Arm Generic Timer defined in the *Arm® Architecture Reference Manual for A-profile architecture*.

This timer provides a single timer frame, without a second view, and without virtual timer capability. REFCLK CNTCTL and REFCLK CNTBase0 are visible in the SCP memory map.

For the SCP memory map, see SCP memory map.

AP_REFCLK generic timers

The Base Block includes two memory-mapped Arm Generic Timers for general-purpose functions. These timers are defined by the *Arm® Architecture Reference Manual for A-profile architecture*. Each timer provides single frames, without a second view, and without virtual timer capability. These timers are called the AP_REFCLK Generic Timers and are in the VSYS.SYSTOP power domain.

In the AP memory map, they are included as the following:

- AP_REFCLK CNTCTL
- AP_REFCLK_S CNTBase1
- AP_REFCLK_NS CNTBase0

Access to the CNTBase0 frame is Non-secure, while the CNTCTL and CNTBase1 frames are Secure.

For the AP memory map, see AP memory map.

Watchdog timers

There are several watchdogs used in Total Compute 2022 Reference Design (RD-TC22) reference subsystems.

The following sections describe the [SCP Watchdog], [Generic Watchdog], and [Trusted Watchdog].

Watchdog security

The SCP watchdog and Trusted Watchdog Timers are accessible by Secure accesses only.

These watchdogs also support halt-on debug functionality, enabling cross-triggers to halt the watchdog.

SCP Watchdog

The SCP Block includes a Cortex®-M System Design Kit Watchdog Timer that protects against lockups in the firmware. This watchdog timer is clocked by REFCLK.

The first time the SCP Watchdog expires, an interrupt to the SCP is generated. If this fails to clear the watchdog, and it expires for a second time, a global reset is generated.

Generic Watchdog

The *Client Base System Architecture Platform Design Document* (PDD) defines and requires a generic watchdog for EL2 software to use.

This watchdog generates the following interrupts:

- The first interrupt is expected to be configured as an EL2 interrupt, and is routed as a *Shared Peripheral Interrupt* (SPI).
- The second interrupt must cause EL2 and higher levels to reset.

The reset of EL2 is supported by routing the second interrupt as an SPI that can be configured as an EL3 interrupt. This is because the application processor cluster in the subsystem implements EL3.

This timer increments by one every REFCLK cycle.

Two memory-mapped register frames manage the Generic Watchdog. In the subsystem, these frames are accessible by Secure and Non-secure accesses.

For more information on the programmers model of the Generic Watchdog, see the section AP memory map.

Trusted Watchdog

An AP includes a Generic Watchdog Timer that protects the Secure Boot process when it is necessary to run untrusted device drivers.

The first time the Trusted Watchdog expires, an interrupt to the CoreLink™ GIC-700 Generic Interrupt Controller is generated. If Secure Boot software fails to clear the watchdog, and it expires for a second time, a global reset is generated.

The Secure Watchdog state is not preserved through reset because this is not a requirement of Client Base System Architecture.

This watchdog increments by one every REFCLK cycle.

4.3.2 Processor core power down considerations

A core timer can generate timer interrupts after entering into *Wait For Interrupt* (WFI) or *Wait for Event* (WFE) mode. So cores can enter WFI and WFE modes without any side effects relating to their timers. However, when a core is powered down, extra steps must be taken because its timer state is lost.

There are two models for powering down a processor core, referred to as the *Application Processor* (AP) wakeup model and *System Control Processor* (SCP) wakeup model.

AP wakeup model

In the AP wakeup model, software running on the APs must ensure that no timers are active on the core that is to be powered down. The core can then be powered down by making a request to the SCP.

The core can later be powered up by another core making a request to the SCP.

SCP wakeup model

The SCP wakeup model applies when powering down the final core.

You must save the timer state to the AP_REFCLK Generic Timers as part of the powerdown sequence. Interrupts are masked during this sequence.

You can then power down the processor by making a request to the SCP. The AP_REFCLK Generic Timers wake the processor when the timer expires.

Before the SCP powers down VSYS.SYSTOP, it must first check whether a wakeup time has been programmed into the AP_REFCLK Generic Timers. If these have been programmed, the SCP must:

1. Save the state of the AP_REFCLK Generic Timers before powering down VSYS.SYSTOP. Other software saves the state of the GIC before VSYS.SYSTOP is powered down.
2. Use the state from the AP_REFCLK Generic Timers to schedule the wakeup of VSYS.SYSTOP at, or slightly before, the time programmed into these timers. To do this, the SCP can use the SCP_REFCLK Generic Timer.
3. After waking VSYS.SYSTOP, the SCP restores state to the AP_REFCLK Generic Timers, and they trigger an interrupt. Other software, as mentioned in the first step, restores state to the GIC. The interrupt, handled by the SPC, causes a wakeup interrupt for the processor.

5. RD-TC22 Fixed Virtual Platform

Arm *Fixed Virtual Platform* (FVP) models are based on Arm Fast Models technology to deliver fast simulations of Arm-based systems. They enable software development ahead of hardware availability and the option to explore the design from a software perspective. Together with a corresponding reference software stack, they enable efficient software and firmware development, reducing the amount of work that is required for development of a complete system.

Arm FVP models a *Programmers' View* (PV) of processors and other devices in a system. A PV is at the level where functional behavior is equivalent to what a programmer would see using the hardware. It sacrifices timing accuracy to achieve fast simulation execution speeds. Therefore, you can use the FVP for confirming software functionality, but must not rely on the accuracy of cycle counts, low-level component interactions, or other hardware-specific behavior. The FVP also cannot be used to measure software performance.

Total Compute FVPs are supplied as standalone executables for Linux. They are not customizable, although some aspects of their behavior can be configured through command-line parameters.

For more information on the Fixed Virtual Platforms, see the following documentation:

- *Fast Models Fixed Virtual Platforms (FVP) Reference Guide*
- *Fast Models Reference Guide*

Reference software stacks are described in RD-TC22 software.

5.1 Reference Configuration

The Total Compute 2022 Reference Design (RD-TC22) FVP models many of the Arm® IP involved in the subsystem, and supports the following configuration.

Reference subsystem configuration for premium smartphones is:

- 8 Armv9.2-A processor cores with *DynamIQ™ Shared Unit* (DSU) connect, configured as:
  - 1 x Cortex®-X4 Core (L1D:64KB, L1I:64KB, L2:2MB)
  - 3 x Cortex®-A720 Core (L1D:32KB, L1I:32KB, L2:512KB)
  - 4 x Cortex®-A520 Core (L1D:32KB, L1I:32KB, L2:256KB per two-core complex)
  - DynamIQ™ Shared Unit-120 (L3:8MB, 4 cache slices)
- Mali™-G720 GPU MC12 (12 GPU cores)
- CoreLink™ CI-700 r2 2x2 mesh with shared 16MB *System Level Cache* (SLC) and 32MB *Snoop Filter* (SF)
- Memory Controller 4x16-bit memory interfaces

- Multiple expansion ports from CoreLink™ NI-700 r2 for NPUs, multimedia components, modem, USB, sensors, and other general peripherals
- Cortex®-M3-based *System Control Processor* (SCP)
- CoreSight™ SoC-600-based debug and profiling

5.2 About Total Compute 2022 Reference Design FVP

The Total Compute 2022 Reference Design (RD-TC22) *Fixed Virtual Platform* (FVP) models premium mobile reference subsystems.

The FVP models the following key components inside the subsystem:

- Arm® Cortex®-X4 Core
- Arm® Cortex®-A720 Core
- Arm® Cortex®-A520 Core
- Arm® Hayden DynamIQ™ Shared Unit
- Arm® Mali™-G720 GPU
- Arm® CoreLink™ CI-700 r2 Coherent Interconnect
- Arm® CoreLink™ NI-700 r2 Network-on-Chip Interconnect
- Arm® CoreLink™ GIC-700 Generic Interrupt Controller
- Arm® CoreLink™ MMU-700 System Memory Management Unit
- *System Control Processor* (SCP) based on the Arm® Cortex®-M3 processor
- Arm® CoreLink™ NIC-450 Network Interconnect
- Arm® CoreLink™ TZC-400 Address Space Controller
- Arm® CoreLink™ PCK-600 Power Control Kit
- On-Chip ROM, RAM, and other peripherals
- Clock generators with support for dynamic clock gating

The FVP does not model every component that is within the reference subsystem. For example, the following subsystem components are not included:

- Arm® CoreSight™ System-on-Chip SoC-600
- Third-party *Dynamic Memory Controller* (DMC)

To provide a complete system boot environment, the FVP also models components that reside outside of the reference subsystem, including DRAM memory and external peripherals. Together, these components are referred to as the Rest of the System.

The FVP is used with the Total Compute reference software stack. See Total Compute on the Arm Developer website for instructions on software, and how to set up and run the FVP.

5.3 Rest of the System

As its name suggests, the *Rest of the System* (RoS) contains the components that do not reside in the Total Compute 2022 Reference Design (RD-TC22) reference subsystem. These components create the necessary environment for running and developing a reference software stack.

The Total Compute 2022 Reference Design (RD-TC22) *Fixed Virtual Platform* (FVP) includes and represents three main parts of a system:

- Reference subsystem or *Compute Subsystem* (CSS)
- *System on Chip* (SoC) which contains the CSS
- Board that contains the SoC

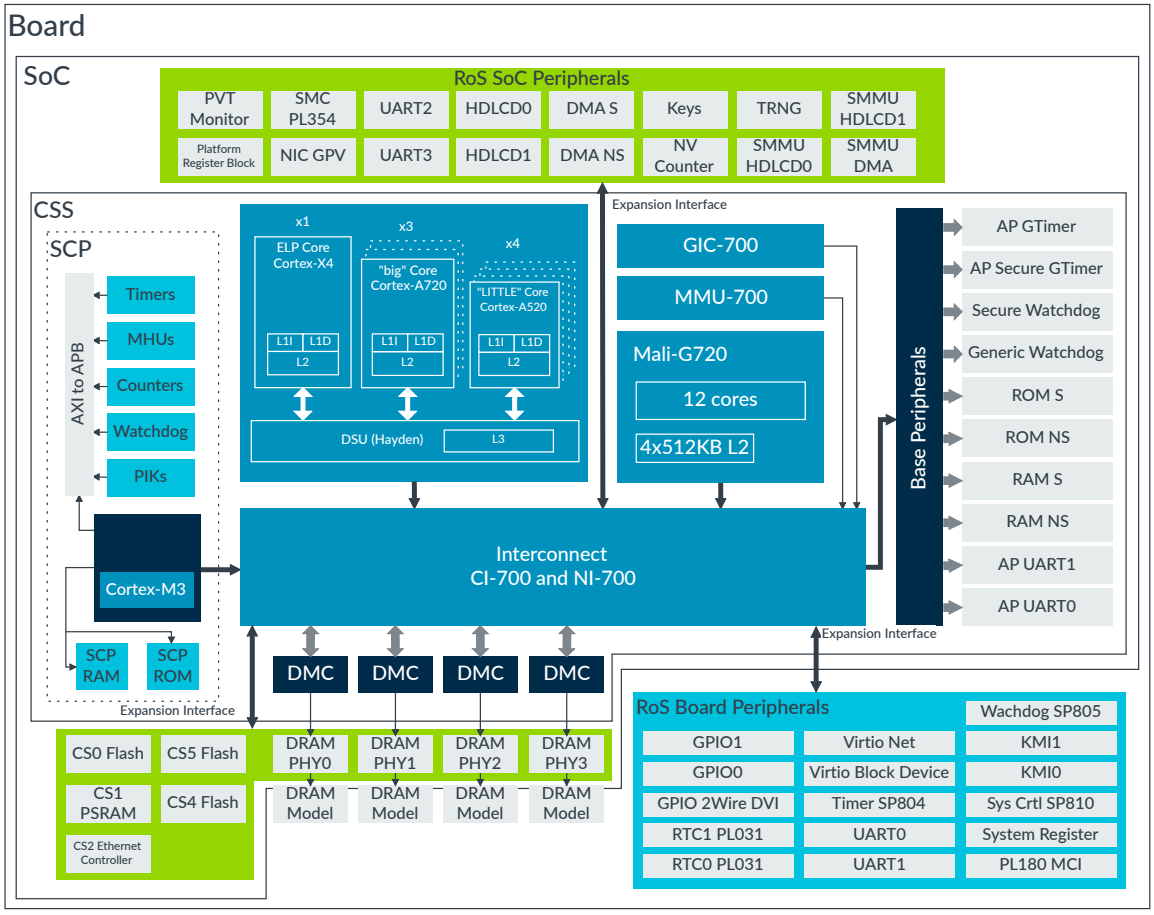
The last two of these three parts are covered by the RoS.

Similar to how the CSS should be integrated in an SoC and a board, the RoS components connect to the CSS through its expansion interfaces. This adds to the expansion sections of the memory map and interrupt map of the CSS.

The following figure shows the block diagram of the RoS.

The block diagram of the RoS is shown in Figure 5-1: RoS block diagram on page 34.

Figure 5-1: RoS block diagram



The following table lists the IP used inside the RoS.

Table 5-1: List of RoS IP

IP name	Function
ADB-400	AMBA domain bridge
NIC-400	Fixed configuration AMBA bus matrix
PL354 SMC	Dual channel combined Static memory / NAND Flash controller
SMC memory	Simple SRAM RTL model
DFI PHY	DFI DDR physical interface
TRNG	Registers for True Random Number Generator
NVCounter	Registers for Non-Volatile Counter
Keys	Register block for Secure keys
PL011 UART	UART for test output
PL180 MCI	Multimedia Card Interface

IP name	Function
PL031 RTC	Real Time clock
SP805 watchdog	Watchdog
SP804 Timer	Dual Timer
PL050 KMI	Keyboard and Mouse Interface
PL061 GPIO	General purpose I/O
PL370 HDLCD	HDLCD controller
PL330 DMA	DMA controller
PSRAM	Pseudo-static DRAM
SMSC 91C111	SMSC 91C111 Ethernet controller

5.3.1 RoS memory map

The RoS memory sections are mapped to the *Compute Subsystem* (CSS) AP memory map Expansion AXI space at the following address ranges:

- `0x000_0800_0000-0x000_1FFF_FFFF`
- `0x000_6000_0000-0x000_7FFF_FFFF`

The following table shows the memory sections and their mapping.

Table 5-2: RoS memory sections

RoS memory section	Beginning address in AP memory map	Size	Description
ROS_SMC_BASE0	`0x000_0800_0000`	64MB	CS0 – Boot flash
ROS_SMC_BASE1	`0x000_0C00_0000`	64MB	CS4 – flash (Non-secure Storage)
ROS_SMC_BASE2	`0x000_1000_0000`	64MB	CS5 – flash (Secure storage)
ROS_SMC_BASE3	`0x000_1400_0000`	64MB	CS1 - PSRAM
ROS_SMC_BASE4	`0x000_1800_0000`	64MB	CS2 – Ethernet controller
ROS_BOARD_BASE	`0x000_1C00_0000`	32MB	RoS board peripherals
PCIE_BASE	`0x000_6000_0000`	512MB	PCIe root complexes
ROS_SOC_BASE	`0x000_7F00_0000`	32MB	RoS SoC peripherals
SDRAM PHY	`0x000_7FB6_0000`	32 x 64KB	SDRAM PHYs (0-31)
SMMU	`0x000_7FB0_0000`	256KB	SMMU

For more information on AP memory, see Memory maps.

RoS board components are memory-mapped to the addresses listed in the following table. The addresses are offset from ROS_BOARD_BASE.

Table 5-3: RoS board peripherals memory map

Start Address Offset	End Address Offset	Size	Component
`0x1F_0000`	`0x1F_FFFF`	64KB	Reserved

Start Address Offset	End Address Offset	Size	Component
0x1E_0000	0x1E_FFFF	64KB	GPIO 1
0x1D_0000	0x1D_FFFF	64KB	GPIO 0
0x19_0000	0x1E_FFFF	256KB	Reserved
0x18_0000	0x18_FFFF	64KB	RTC 1
0x17_0000	0x17_FFFF	64KB	RTC 0
0x16_0000	0x16_FFFF	64KB	GPIO 2 Wire (DVI)
0x15_0000	0x15_FFFF	64KB	Virtio Net
0x14_0000	0x14_FFFF	64KB	Virtio RNG
0x13_0000	0x13_FFFF	64KB	Virtio Block Device
0x12_0000	0x12_FFFF	64KB	Reserved
0x11_0000	0x11_FFFF	64KB	SP084 Dual Timer
0x10_0000	0x10_FFFF	64KB	Reserved
0x0F_0000	0x0F_FFFF	64KB	SP805 Watchdog
0x0B_0000	0x0E_FFFF	256KB	Reserved
0x0A_0000	0x0A_FFFF	64KB	UART 3
0x09_0000	0x09_FFFF	64KB	UART 2
0x08_0000	0x08_FFFF	64KB	Reserved
0x07_0000	0x07_FFFF	64KB	KMI 1
0x06_0000	0x06_FFFF	64KB	KMI 0
0x05_0000	0x05_FFFF	64KB	PL180 MCI
0x03_0000	0x04_FFFF	128KB	Reserved
0x02_0000	0x02_FFFF	64KB	SP810 Sysctrl
0x01_0000	0x01_FFFF	64KB	System registers
0x00_0000	0x00_FFFF	64KB	Reserved

RoS SoC components are memory-mapped to the addresses listed in the following table. The addresses are offset from ROS_SOC_BASE.

Table 5-4: RoS SoC peripherals memory map

Start Address Offset	End Address Offset	Size	Name
0xFF_0000	0xFF_FFFF	64KB	SoC SOR HDLCD security override
0xFE_0000	0xFE_FFFF	64KB	Platform register block
0xFD_0000	0xFD_FFFF	64KB	SMC PL354 Cfg
0xF9_0000	0xFC_FFFF	256KB	Reserved
0xF8_0000	0xF8_FFFF	64KB	UART 1
0xF7_0000	0xF7_FFFF	64KB	UART 0
0xF6_0000	0xF6_FFFF	64KB	HDLCD 0
0xF5_0000	0xF5_FFFF	64KB	HDLCD 1
0xF3_0000	0xF3_FFFF	64KB	PCIe Root port
0xF2_0000	0xF2_FFFF	64KB	PCIe Macro
0xF1_0000	0xF1_FFFF	64KB	DMA NS

Start Address Offset	End Address Offset	Size	Name
0xF0_0000	0xF0_FFFF	64KB	DMA S
0xE9_1000	0xEF_FFFF	444KB	Reserved
0xE9_0000	0xE9_0FFF	4KB	GPIO
0xE8_0000	0xE8_FFFF	64KB	Keys
0xE7_0000	0xE7_FFFF	64KB	NV Counter
0xE6_0000	0xE6_FFFF	64KB	TRNG
0xE5_0000	0xE5_0FFF	4KB	Surge
0xE4_0000	0xE4_0FFF	4KB	PVT Monitor CPU 1
0xE3_0000	0xE3_0FFF	4KB	PVT Monitor CPU 0
0xE2_0000	0xE2_0FFF	4KB	PVT Monitor GPU
0xE1_0000	0xE1_0FFF	4KB	PVT Monitor SoC
0xE0_0000	0xE0_0FFF	4KB	PVT Monitor STD
0xD0_0000	0xDF_FFFF	1MB	NIC GPV
0xB0_0000	0xCF_FFFF	32 x 64KB	SDRAM PHY 0-31 (not used)
0x04_0000	0xAF_FFFF	11008KB	Reserved
0x03_0000	0x03_FFFF	64KB	SMMU_USB
0x02_0000	0x02_FFFF	64KB	SMMU_HDLCD 1
0x01_0000	0x01_FFFF	64KB	SMMU_HDLCD 0
0x00_0000	0x00_FFFF	64KB	SMMU_DMA

5.3.2 RoS interrupt map

The interrupts of the RoS components are mapped to the AP expansion interrupts area of the AP interrupt map, starting at interrupt ID 128. The following table summarizes these interrupts, where

- Interrupt ID = 128 + INTERRUPT_OFFSET
- GIC IRQ NUM = Interrupt ID - 32

Table 5-5: RoS interrupt map

INTERRUPT_OFFSET	Interrupt source
0-2	Reserved
3	RTC1
4	RTC0 (EXT_IRQ[0])
5	UART0 (EXT_IRQ[1]) (Board)
6	UART1 (EXT_IRQ[2]) (Board)
7	KMI1
8	GPIO0
9	GPIO1
10	I2C GPIO
11	MCIINTR0

INTERRUPT_OFFSET	Interrupt source
12	MCIINTR1
13	SMSC 91C111
14-18	Reserved
19	UART0 (SoC)
20	UART1 (SoC)
21	HDLCD controller 0
22	SMC PL354 Interface 0
23	SMC PL354 Interface 1
24-28	Reserved
29	HDLCD controller 1
30	SMMU Combined Secure Interrupt
31	SMMU Combined Non-secure Interrupt
32	Reserved
33	Reserved
34-41	DMA0 IRQ7-0
42	DMA0 IRQ ABORT
43	TRNG
44-51	DMA1 IRQ7-0
52	DMA1 IRQ ABORT
53-73	Reserved
108	Virtio Block Device
110	Virtio Rng
109	Virtio net
99	RTCC
100	WDT
101	KMI0
102	Dual Timer
103	System registers
104	System register – USB
105	System register – Tile
106	System register – Push button
107	System register – Ethernet

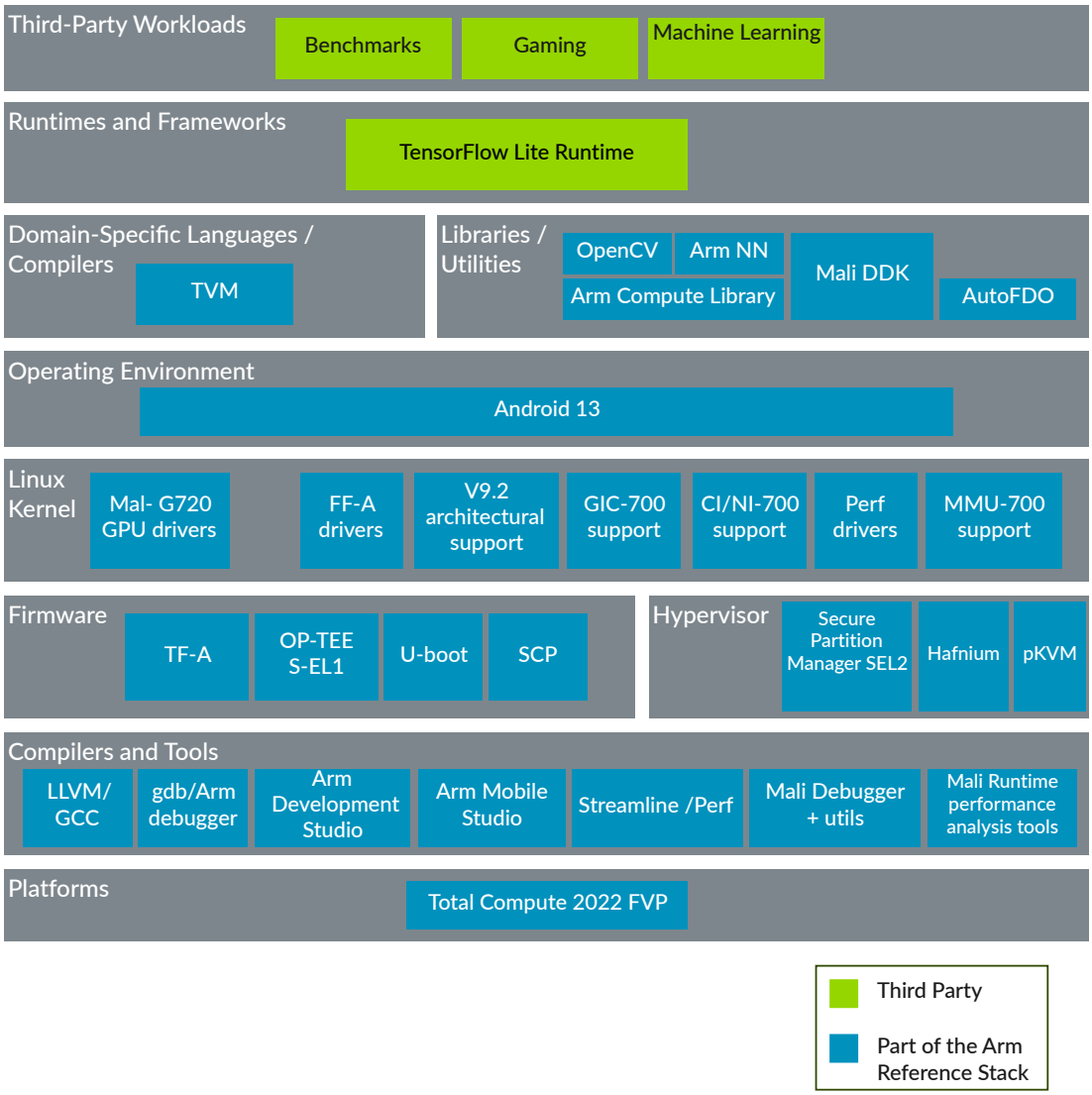
6. RD-TC22 software

The Total Compute reference software stack is a fully integrated open-source software stack, from Firmware up to Android™. This stack enables pre-silicon software development. It provides a starting point to modify, extend, and develop the software for a *System on Chip* (SoC) based on the Total Compute reference designs. By using it, it allows early testing, system integration, and validation, reducing time to market and boosting product security and reliability.

The software stack includes open-source code available from the relevant upstream projects, including *System Control Processor* (SCP) firmware, Trusted firmware, Linux kernel, Android, Arm NN, and many more.

Figure 6-1: Components of Total Compute 2022 software stack on page 40 shows all the components of the TC2022 reference software stack. The main components are described in Software components.

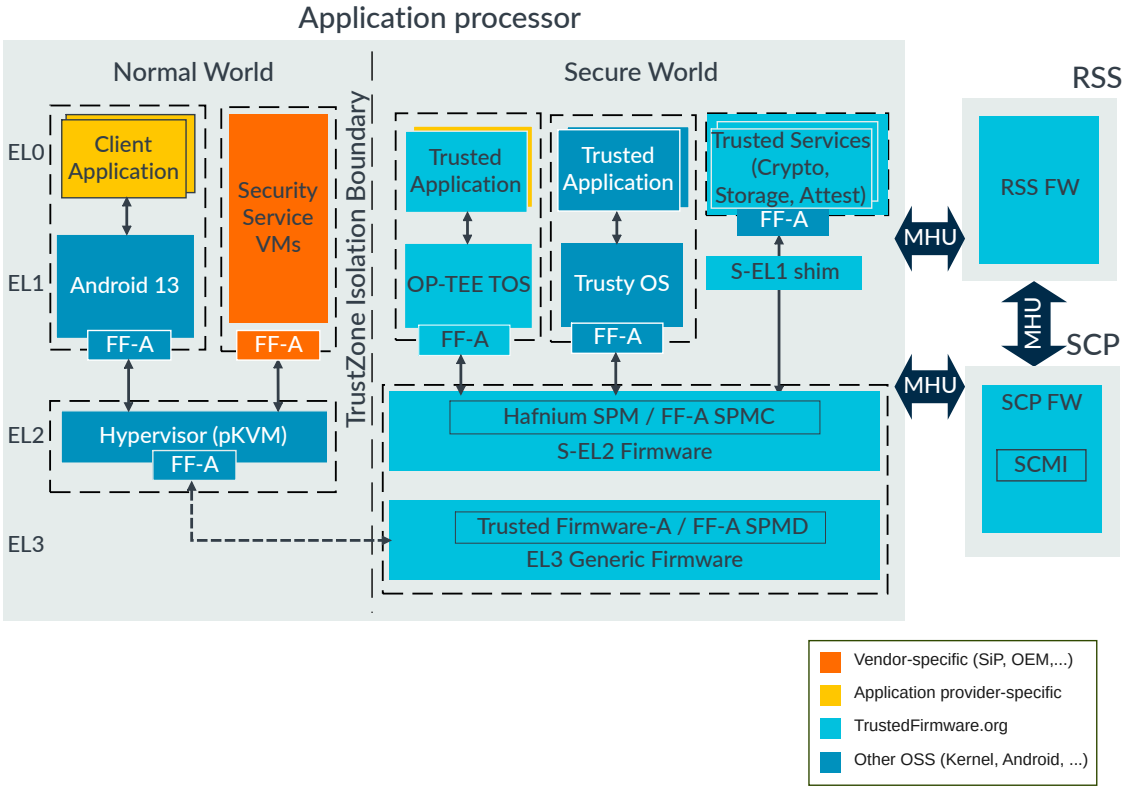
Figure 6-1: Components of Total Compute 2022 software stack



The software stack is developed and validated against the Total Compute FVP and proven on Arm® internal hardware emulation environments. See RD-TC22 Fixed Virtual Platform.

For a runtime view of the software stack, see Figure 6-2: Runtime view of Total Compute 2022 software stack on page 41.

Figure 6-2: Runtime view of Total Compute 2022 software stack



For more information on the software stack, instructions on how to set it up, and how to run it on the FVP, see Total Compute on the Arm Developer website.

6.1 Software components

Main components of the reference software stack include firmware, the Kernel, and file system components.

These components are described as follows:

SCP firmware, see SCP firmware

The *System Control Processor* (SCP) firmware performs low-level system initialization at boot-up, and abstracts power and system management tasks away from *Application Processors* (APs). It complies with the Arm® *System Control and Management Interface* (SCMI) specification. The firmware consists of:

- SCP-BL1: SCP boot ROM
- SCP-BL2: SCP runtime firmware

AP firmware, see AP firmware

An AP is required to boot the reference design up to the point where the *Rich Operating System* (ROS) execution starts. The AP firmware comprises the following components:

- Arm *Trusted Firmware-A* (TF-A) BL1: AP trusted ROM
- Arm TF-A BL2: AP trusted boot firmware
- Arm TF-A BL31: AP EL3 Runtime firmware
- Arm TF-A BL32: AP secure-EL1 Payload
- BL33: AP Normal world firmware
  - U-Boot or *Unified Extensible Firmware Interface* (UEFI), see U-Boot, performs hardware initialization and acts as the boot loader for rich operating systems like Linux Kernel.

Kernel, see Linux Kernel

The Total Compute 2022 software stack uses the Android™ common Kernel. The Linux Kernel supports features that are required to develop and demonstrate the RD-TC22 reference subsystem-specific hardware features, including:

- *Message Handling Unit* (MHU)
- Arm architectural features
- ***Pointer Authentication Code*** **(PAC)**

  Protects against return-oriented programming attacks. At the start of a function, the return address in the *Link Register* (LR) is signed. This means that a PAC is added to the upper order bits of the register. Before returning, the return address is authenticated using the PAC. If the check fails, an exception is generated when the address is used for a branch.
- ***Memory Tagging Extension*** **(MTE)**

  Designed to detect memory safety violations and to increase robustness against attacks that such violations enable. MTE implements lock and key access to the memory. Memory locations are tagged by adding four bits of metadata to each 16 bytes of physical memory.
- ***Branch Target Identification*** **(BTI)**

  Used to guard against the execution of instructions that are not the intended target of a branch.

Android, see Android

Total Compute 2022 software stack runs Android™ mobile operating systems.

6.1.1 SCP firmware

The *System Control Processor* (SCP) firmware runs on the Cortex®-M3 processor of the SCP Block that is in the always-on power domain AONTOP. It is responsible for low-level system management.

The SCP firmware performs low-level hardware initialization, and manages the overall power, clock, reset, and system control of the subsystem. The SCP firmware is an inherently trusted part of the

software. To prevent tampering, internal private RAM is used for execution and storage. APs cannot directly access the SCP memory map, rather, it communicates with the SCP through the MHU.

SCP firmware supports:

- Powerup sequence and system startup
- Initialization of hardware configurations
- *Phase-Locked Loops* (PLLs) and clock management
- Servicing of power state requests from *OS Power Management* (OSPM) Software

SCP firmware contains two components: SCP boot ROM and SCP runtime firmware.

For more information on SCP firmware, see An Overview of the opensource Arm System Control Processor (SCP) Firmware Architecture.

6.1.1.1 SCP boot ROM

The SCP boot ROM code executes after a Cold reset.

It performs the following functions:

- Set up the generic timer, *Universal Asynchronous Receiver/Transmitter* (UART) console, and clocks
- Power up primary *Application Processor* (AP), normally CPU0.
- Copy and authenticate SCP runtime firmware from Secure RAM to SCP private RAM after *Trusted Firmware for A-profile* (TF-A) BL2 has loaded the image from *Firmware Image Package* (FIP) to the Secure RAM.

6.1.1.2 SCP runtime firmware

The SCP runtime code executes after it is copied into SCP private RAM.

It performs the following functions:

- Set up the SCP peripherals
- Configure the system controllers, including the memory controller and coherent interconnect
- Respond to *System Control and Management Interface* (SCMI) messages through *Message Handling Unit* (MHU) version 2.0 for processor power control and *Dynamic Voltage and Frequency Scaling* (DVFS)
- Voltage and power domain management
- Clock management

6.1.2 AP firmware

The *Application Processor* (AP) firmware consists of the code that is required to boot the reference design up to the point where the *Rich Operating System* (ROS) execution starts. This firmware

performs architecture and platform initialization. It also loads and initializes Secure world images like the Secure partition manager and Trusted OS. AP firmware is largely built based on Trusted Firmware. Its main components are described in the following sections.

For more information on Trusted Firmware-A, see Trusted Firmware-A Documentation.

6.1.2.1 Trusted firmware-A BL1

Trusted firmware-A BL1 is located in the Trusted ROM of the subsystem. It is the first code the AP executes after it comes out of reset. It runs at EL3.

Its main functions are listed following:

- Minimal architectural initialization, like exception vectors and processor initialization
- Minimal platform initialization, like Trusted Watchdog and *Memory Management Unit* (MMU)
- Load and authenticate firmware-A BL2 from *Firmware Image Package* (FIP) into Secure RAM located in Base Block.

When these functions are done, BL1 passes control to BL2.

6.1.2.2 Trusted firmware-A BL2

Trusted firmware BL2 runs at S-EL1. It performs any additional architectural initialization required for subsequent stages of TF-A and Normal world software.

The AP Trusted firmware BL2 configures the TrustZone Controller and carves out a memory region in DRAM for Secure and Normal world use. It also enables platform security features.

Trusted firmware BL2 Loads and authenticates the following images from FIP into Secure RAM:

- *System Control Processor* (SCP), BL2 image
- AP EL3 Runtime firmware, BL31 image
- AP Secure Payload, BL32 image
- AP Normal world firmware - U-boot, BL33 image

When it is done, BL2 passes control to BL31.

6.1.2.3 Trusted firmware BL31 and BL32

Trusted firmware BL31 runs in Secure SRAM and at EL3.

Its primary purpose is to handle transitions between the Normal and Secure world. It provides the following runtime services:

***Power State Coordination Interface* (PSCI)**

The AP Trusted firmware BL31 defines a *Secure Monitor Call* (SMC) interface to support Rich OS Power Management in accordance to the Power PSCI System Software on Arm Systems. The Linux Processor idle framework uses PSCI to power up or power down the AP cores.

Secure Monitor framework

Handles all SMCs in the AP Trusted RAM firmware. The framework handles the transition to Trusted World execution and distributes the SMCs to the correct SMC handler.

***Secure Partition Manager Dispatcher* (SPMD)**

RD-TC22 enables SPMD as the Secure payload dispatcher. The SPMD relays the *Firmware Framework for A-profile* (FF-A) messages from Normal to Secure world or the opposite way around. This replaces the Trusted OS-specific dispatcher from EL3 runtime firmware.

Trusted firmware BL32 contains security payload, including:

***Secure Partition Manager* (SPM)**

RD-TC22 enables the FEAT S-EL2 architectural extension, using Hafnium as the *Secure Partition Manager Core* (SPMC). The BL32 option in TF-A is repurposed to specify the SPMC image. The SPMC component runs at the S-EL2 Exception level.

Secure Partitions

A Software image that is isolated using SPM is a Secure Partition. RD-TC22 enables *Open Portable Trusted Execution Environment* (OP-TEE) and Trusted services like crypto and Secure storage as Secure Partitions.

OP-TEE

The OP-TEE Trusted OS is virtualized using Hafnium at S-EL2. The OP-TEE OS for Total Compute is built with FF-A and SEL2 SPMC support. This enables OP-TEE as a Secure Partition running in an isolated address space managed by Hafnium. The OP-TEE Kernel runs at S-EL1 with Trusted applications running at S-EL0.

Trusty TEE

Trusty is a secure OS that provides a *Trusted Execution Environment* (TEE). Trusty is virtualized using Hafnium at S-EL2. Trusty for Total Compute is built with FF-A and S-EL2 SPMC support. This enables Trusty as a Secure Partition running in an isolated address space managed by Hafnium. The Trusty Kernel runs at S-EL1 with Trusted applications running at S-EL0.

Trusted Services

Trusted services like the crypto service and Secure storage run as S-EL0 Secure Partitions using a Shim layer at S-EL1. The crypto service and the S-EL1 Shim layer are built as a single image. The Shim layer forwards FF-A calls from S-EL0 to S-EL2.

6.1.2.4 U-Boot

TF-A BL31 passes execution control to the U-boot bootloader, BL33.

U-boot in RD-TC22 supports the following image formats:

FitImage format

Contains the Linux Kernel and Busybox RAM disk, which are authenticated and loaded in their respective positions in DRAM. Execution is then handed off to the kernel.

Android boot image

Contains the Linux Kernel and Android™ RAM disk.

If using *Android Verified Boot* (AVB), the boot.img file is loaded from the *MultiMediaCard* (MMC) to DRAM, authenticated, and then execution is handed off to the Kernel. The boot.img file is one of the files that is created when the Total Compute Software Stack is built.

6.1.3 Linux Kernel

The Linux Kernel in Total Compute 2022 Reference Design (RD-TC22) contains subsystem-specific features that demonstrate the capabilities of the reference subsystem.

Apart from the default configuration, it enables the following:

- Arm *Message Handling Unit* (MHU) version 2.0 controller driver
- Arm TF-A driver
- *Open Portable Trusted Execution Environment* (OP-TEE) driver with TF-A transport support
- Arm TF-A user-space interface driver

6.1.4 Android

Total Compute 2022 Reference Design (RD-TC22) has support for the *Android™ Open-Source Project* (AOSP). AOSP contains the Android framework, native libraries, Android runtime, and the *Hardware Abstraction Layers* (HALs) for the Android operating system.

The RD-TC22 device profile defines the required variables for Android such as partition size and product packages, and has support for the following configuration of Android:

Software rendering

This profile has support for Android UI, and boots Android to the home screen. It uses SwiftShader to bring up the Android home screen. The SwiftShader is a processor-based implementation of the Vulkan graphics API by Google.

Hardware rendering

This profile supports android booting to homescreen using Mali-G720, that is part of the FVP model. Both GLES3.2 and Vulkan APIs are supported.

6.1.5 Machine Learning applications

To validate the Machine Learning (ML)-related compute functionality, Total Compute 2022 reference software adds some basic sample applications to execute user-defined ML workloads

using Google's TensorFlow™ Lite library. For more information, see the Total Compute User Guide from Total Compute on the Arm Developer website.

The current application allows developers to experiment with executing workloads on the CPU cores, and the future revisions will allow GPU cores to be exercised.

7. Programmers model

The Programmers model describes the Total Compute 2022 Reference Design (RD-TC22) subsystem memory maps, interrupt maps, and register definitions.

7.1 Memory maps

The *System Control Processor* (SCP) and *Application Processor* (AP) have their own memory maps which are different from each other. These memory maps are detailed following.

7.1.1 AP memory map

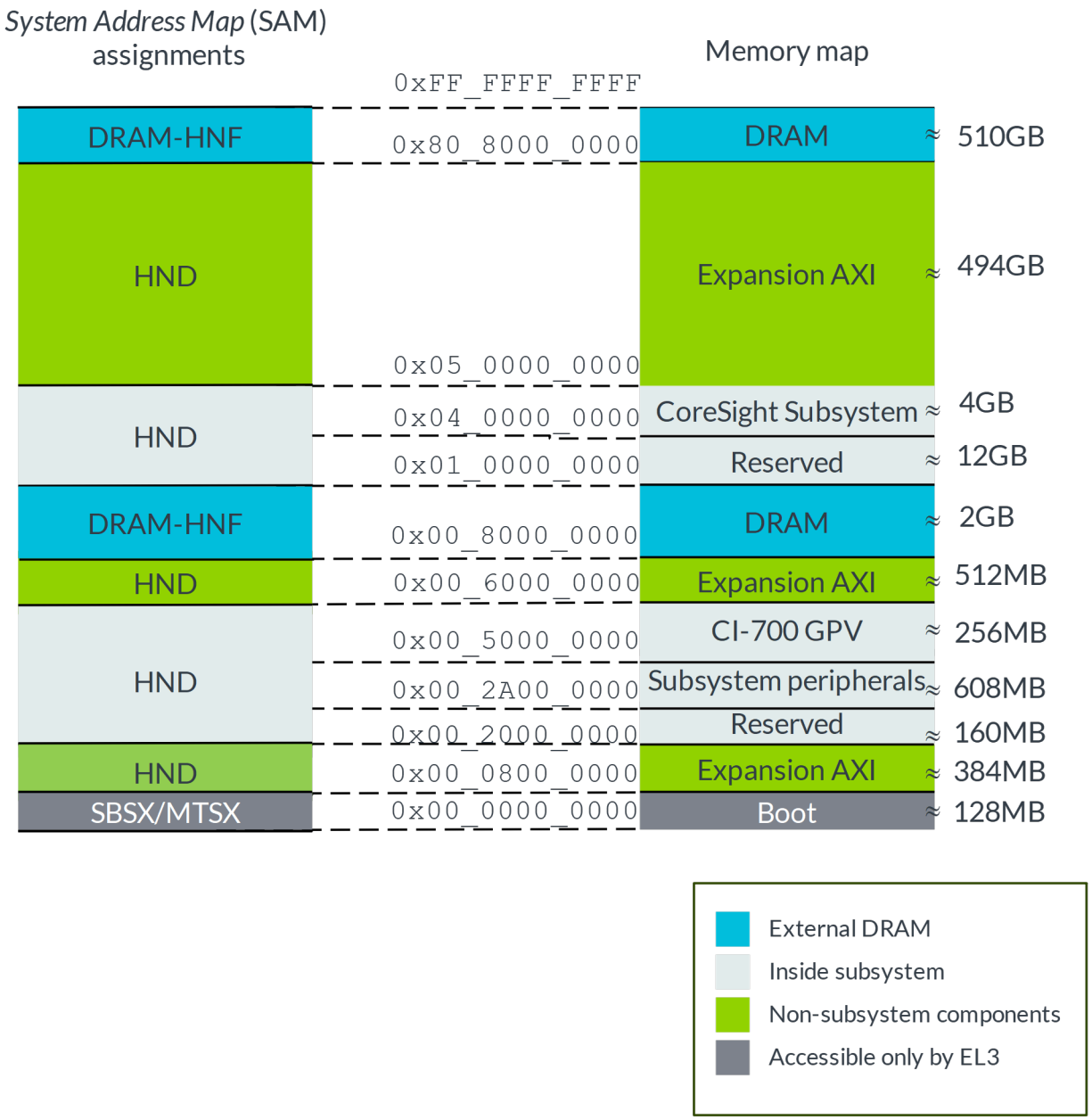
This subsection describes the memory map for the *Application Processor* (AP).

The AP memory map is visible to the following:

- *Application Processors* (APs)
- *System Control Processor* (SCP)
- Embedded Trace Router
- Coherent Expansion *Advanced eXtensible Interface* (AXI) Responder interfaces
- Application Processor *Debug Access Port* (DAP)

The following figure shows a top-level representation of the AP memory map.

Figure 7-1: Application processor memory map



These security attributes are split into the following regions:

Always Secure access

A region that is only accessible to Secure transactions. Any Non-secure access targeting these, results in a DECERR decode error response.

Secure and Non-secure access

A region that is accessible to both Secure and Non-secure transactions.

Programmable access security

Programmable access security is also known as Securable. It is a region that is defined to be independently software-configurable, and can be changed between the following states by trusted software:

- Always Secure access
- Secure and Non-secure access

These can be configured in the NI-700 or in the component itself, and the default state is Secure access only from reset.

User-defined

A region that is mapped to expansion interfaces. Their access security is defined by the components outside of the Arm Total Compute 2022 Reference Design (RD-TC22) subsystem. These components must use the ARPROT[1] or AWPROT[1] bits provided on the expansion interfaces to determine the security permission of each access. Any accesses that fail any external security checks must result in a DECERR response.

In general, unless explicitly stated otherwise:

- When a region maps a peripheral or device that occupies less than the region size used, access to the unmapped region results in a DECERR response. For example, when a peripheral occupies 4KB from the 64KB region that is reserved for it.
- Accesses to reserved areas within the memory map also result in a DECERR response. When accessing areas that peripherals or devices occupy, the peripherals or devices themselves determine the response to return. These areas can include unmapped or reserved areas within the areas that the peripheral or device occupies.
- If the SCP needs to access external peripherals that reside outside the subsystem, those peripherals should be mapped to the lower 2GB of the address space.

The following table lists the memory map for the AP.

Table 7-1: AP memory map

Start address	End address	Region	Additional information
`0x00_0000_0000`	`0x00_03FF_FFFF`	Secure Boot ROM	Accesses above 512KB (`0x0007_FFFF`) will result in a decode error. For accesses above the configured size but below 512KB, reads will return zero. In RD-TC22 FVP this ROM is configured to be 512KB.
`0x00_0400_0000`	`0x00_04FF_FFFF`	Secure RAM	Accesses above 512KB (`0x0407_FFFF`) will result in a decode error. For accesses above the configured size but below 512KB, writes will be ignored and reads will return zero. In RD-TC22 FVP this RAM is configured to be 512KB.

Start address	End address	Region	Additional information
0x00_0500_0000	0x00_05FF_FFFF	Non-secure ROM	Accesses above 512KB (0x0507_FFFF) will result in a decode error. For accesses above the configured size but below 512KB, reads will return zero. In RD-TC22 FVP this ROM is configured to be 512KB.
0x00_0600_0000	0x00_06FF_FFFF	Non-secure RAM	Accesses above 512KB (0x0607_FFFF) will result in a decode error. For accesses above the configured size but below 512KB, writes will be ignored and reads will return zero. In RD-TC22 FVP this RAM is configured to be 512KB.
0x00_0800_0000	0x00_1FFF_FFFF	Expansion *Advanced eXtensible Interface* (AXI) (Peripheral 0)	-
0x00_2000_0000	0x00_20FF_FFFF	Reserved	-
0x00_2100_0000	0x00_21FF_FFFF	DMC0 configuration	-
0x00_2200_0000	0x00_22FF_FFFF	DMC1 configuration	-
0x00_2300_0000	0x00_23FF_FFFF	DMC2 configuration	Reserved for C2 Mobile
0x00_2400_0000	0x00_24FF_FFFF	DMC3 configuration	Reserved for C2 Mobile
0x00_2500_0000	0x00_25FF_FFFF	TZC400-0 configuration	-
0x00_2600_0000	0x00_26FF_FFFF	TZC400-1 configuration	-
0x00_2700_0000	0x00_27FF_FFFF	TZC400-2 configuration	Reserved for C2 Mobile
0x00_2800_0000	0x00_28FF_FFFF	TZC400-3 configuration	Reserved for C2 Mobile
0x00_2900_0000	0x00_2A3F_FFFF	Reserved	-
0x00_2A40_0000	0x00_2A40_FFFF	Non-secure Universal Asynchronous Receiver/Transmitter (UART)	-
0x00_2A41_0000	0x00_2A41_FFFF	Secure UART	-
0x00_2A42_0000	0x00_2A42_FFFF	Reserved	-
0x00_2A43_0000	0x00_2A43_FFFF	REFCLK CNTControl	-
0x00_2A44_0000	0x00_2A44_FFFF	Generic Watchdog Control	-
0x00_2A45_0000	0x00_2A45_FFFF	Generic Watchdog Refresh	-
0x00_2A46_0000	0x00_2A47_FFFF	Reserved	-
0x00_2A48_0000	0x00_2A48_FFFF	Trusted Watchdog Control	-
0x00_2A49_0000	0x00_2A49_FFFF	Trusted Watchdog Refresh	-
0x00_2A4A_0000	0x00_2A4A_FFFF	System ID registers	-
0x00_2A4B_0000	0x00_2A7F_FFFF	Reserved	-
0x00_2A80_0000	0x00_2A80_FFFF	REFCLK CNTRead	-
0x00_2A81_0000	0x00_2A81_FFFF	AP_REFCLK_CNTCTL	-
0x00_2A82_0000	0x00_2A82_FFFF	AP_REFCLK_S_CNTBase0	-
0x00_2A83_0000	0x00_2A83_FFFF	AP_REFCLK_NS_CNTBase1	-
0x00_2A84_0000	0x00_2B7F_FFFF	Reserved	-

Start address	End address	Region	Additional information
0x00_2B80_0000	0x00_2B81_FFFF	Reserved	-
0x00_2B82_0000	0x00_2CBF_FFFF	Reserved	-
0x00_2CC0_0000	0x00_2CEF_FFFF	Display block	-
0x00_2CF0_0000	0x00_2CFF_FFFF	Reserved	-
0x00_2D00_0000	0x00_2DFF_FFFF	GPU block	-
0x00_2E00_0000	0x00_2FFF_FFFF	Reserved	-
0x00_3000_0000	0x00_37FF_FFFF	CoreLink™ GIC-700 Generic Interrupt Controller	-
0x00_3800_0000	0x00_3EFF_FFFF	Reserved	-
0x00_3F00_0000	0x00_43FF_FFFF	CoreLink™ MMU-700 System Memory Management Unit	-
0x00_4400_0000	0x00_44FF_FFFF	Reserved	-
0x00_4500_0000	0x00_4500_FFFF	AP to SCP Non-secure MHU	-
0x00_4501_0000	0x00_4501_FFFF	SCP to AP Non-secure MHU	-
0x00_4502_0000	0x00_453F_FFFF	Reserved	-
0x00_4540_0000	0x00_4540_FFFF	AP to SCP Secure MHU	-
0x00_4541_0000	0x00_4541_FFFF	SCP to AP Secure MHU	-
0x00_4542_0000	0x00_4EFF_FFFF	Reserved	-
0x00_4F00_0000	0x00_4FFF_FFFF	NI-700 GPV	-
0x00_5000_0000	0x00_5FFF_FFFF	CI-700 GPV	-
0x00_6000_0000	0x00_67FF_FFFF	Expansion AXI (PCIe)	-
0x00_6800_0000	0x00_6FFF_FFFF	Reserved	-
0x00_7000_0000	0x00_7FFF_FFFF	Expansion AXI (Peripheral 0)	-
0x00_8000_0000	0x00_FFFF_FFFF	DRAM	-
0x01_0000_0000	0x01_007F_FFFF	Cluster0 Utility space	-
0x01_0080_0000	0x03_FFFF_FFFF	Reserved	-
0x04_0000_0000	0x04_3FFF_FFFF	Debug memory map	-
0x04_4000_0000	0x05_FFFF_FFFF	Expansion AXI (Peripheral 1)	-
0x06_0000_0000	0x06_7FFF_FFFF	Reserved	-
0x06_8000_0000	0x06_FFFF_FFFF	Reserved	-
0x07_0000_0000	0x07_7FFF_FFFF	Reserved	-
0x07_8000_0000	0x07_FFFF_FFFF	Reserved	-
0x08_0000_0000	0x0F_FFFF_FFFF	Reserved	-
0x10_0000_0000	0x10_000F_FFFF	Expansion peripheral	-
0x10_0010_0000	0x10_001F_FFFF	ISP (expansion)	-
0x10_0020_0000	0x10_002F_FFFF	Sensor Hub (expansion)	-
0x10_0030_0000	0x80_7FFF_FFFF	Expansion AXI (Peripheral 1)	-
0x80_8000_0000	0xFF_FFFF_FFFF	DRAM	-

7.1.2 SCP memory map

The *System Control Processor* (SCP) is a Cortex®-M3-based subsystem which implements a 32-bit address space. The Cortex®-M3 uses a fixed high-level memory map as specified in the *Arm®v7-M Architecture Reference Manual*.

The SCP is a Trusted CPU and will always run Trusted code. Therefore, all the regions of the SCP memory map that are not mapped to the *Application Processor* (AP) Memory Map are Secure by default.

A boot ROM and an on-chip SRAM are mapped in the bottom 512MB of the address space. The first 1MB of the top 512MB of address space is reserved for the *Private Peripheral Bus* (PPB).

This region is further divided into the following spaces:

Internal PPB

The Internal PPB space is the bottom 256KB of the PPB space and is accessed through an *Advanced High-performance Bus Lite* (AHB-Lite) bus with the SCP subsystem. The following Cortex-M3 system components are in this space:

- *System Control Space* (SCS)
- *Flash Patch and Breakpoint* (FPB)
- *Data Warehouse Trace* (DWT)
- *Instrumentation Trace Macrocell* (ITM)

For more information on Cortex-M3 address space, see the *Arm® Cortex®-M3 Processor Technical Reference Manual*.

External PPB

The External PPB space contains more Cortex-M3 system components. These are generally debug-related components like such as the SWO, *Embedded Trace Macrocell* (ETM), *Cross Trigger Interface* (CTI), SCP Funnel, and ROM Table. The rest of the address space is Reserved. The ROM table follows the format specified in the *Arm® Cortex®-M3 Processor Technical Reference Manual*, with the part number set to `0x4D0`.

For more information on Cortex-M3 address space, see the *Arm® Cortex®-M3 Processor Technical Reference Manual*.

All other address space is accessed through the Cortex-M3 system bus, and is divided into the following regions:

SCP Peripherals (64KB)

The SCP Peripherals region contains the SCP peripherals such as Generic Timers, Generic Counters, Watchdog Timer, Configuration registers, and Power registers.

External RAM (1GB)

The External RAM region is assumed to behave as Normal memory. For information on the Normal memory type, see the *Arm®v7-M Architecture Reference Manual*.

This region is mapped to a single contiguous 1GB region of the AP memory map starting at `0x00_4000_0000`. Any SCP memory accesses in the External RAM region, `0x6000_0000` to `0x9FFF_FFFF`, targets memory locations in the region `0x00_4000_0000` to `0x00_7FFF_FFFF` of the AP memory map.

This region of the AP memory map contains an expansion *Advanced eXtensible Interface* (AXI) space.

External Device (1GB)

The External Device region is intended for off-chip Device memory. For information on the Device type memory, see the *Arm®v7-M Architecture Reference Manual*.

This region is mapped to a single contiguous 1GB region of the AP memory map starting at `0x00_0000_0000`. Any SCP memory accesses in the External Device region, `0xA000_0000` to `0xDFFF_FFFF`, targets memory locations in the region `0x00_0000_0000` to `0x00_3FFF_FFFF` of the AP memory map.

This region of the AP memory map contains the Boot area, Total Compute 2022 Reference Design (RD-TC22) system peripherals, and an expansion AXI area.

Expansion AHB

The Expansion AHB region contains:

- 511MB, starting at `0x00_E010_0000`
- 64MB, starting at `0x00_4000_0000`

All vendor-specific peripherals such as the register state for controlling the *Power Management Integrated Circuit* (PMIC), *Phase-Locked Loops* (PLLs) and *Process, Voltage, and Temperature* (PVT) sensors can be put in this space. Peripheral I/O control such as the *Serial Peripheral Interface* (SPI), *Integrated Circuit* (I2C) can also be put in this space.

This area is accessible through the SCP external AHB Expansion interface. For more information on memory types such as Device and Normal, and other recommendations regarding the use of these memory areas, see the *Arm®v7-M Architecture Reference Manual*.

Reserved

All remaining regions are Reserved. Any access targeting these regions results in a BusFault exception.

The following table lists the memory map of the System Control Processor.

Table 7-2: SCP memory map

Start address	End address	Region	Additional information
`0x0000_0000`	`0x0FFF_FFFF`	Boot ROM	Access above 512KB (`0x0007_FFFF`) will result in a decode error. Above the configured size but below 512KB, reads will return zero
`0x1000_0000`	`0x1FFF_FFFF`	SRAM	Access above 512KB (`0x0407_FFFF`) will result in a decode error. Above the configured size but below 512KB, writes will be ignored and reads will return zero
`0x2000_0000`	`0x3FFF_FFFF`	Reserved	-

Start address	End address	Region	Additional information
0x4000_0000	0x43FF_FFFF	SCP AHB Expansion	-
0x4400_0000	0x4400_0FFF	REFCLK CNTCTL	-
0x4400_1000	0x4400_1FFF	REFCLK CNTBase0	-
0x4400_2000	0x4400_2FFF	AP to SCP Non-secure MHU	-
0x4400_3000	0x4400_3FFF	SCP to AP Non-secure MHU	-
0x4400_4000	0x4400_4FFF	AP to SCP Secure MHU	-
0x4400_5000	0x4400_5FFF	SCP to AP Secure MHU	-
0x4400_6000	0x4400_6FFF	Watchdog (CMSDK)	-
0x4400_7000	0x4400_9FFF	Reserved	-
0x4400_A000	0x4400_5FFF	CS CNTControl	-
0x4401_0000	0x4FFF_FFFF	Reserved	-
0x5000_0000	0x5000_FFFF	SCP PIK	-
0x5001_0000	0x5001_FFFF	Reserved	-
0x5002_0000	0x5002_FFFF	Debug PIK	-
0x5003_0000	0x5003_FFFF	Debug sensor Group	-
0x5004_0000	0x5004_FFFF	System PIK	-
0x5005_0000	0x5005_FFFF	System sensor Group	-
0x5006_0000	0x5006_FFFF	CPU PIK	-
0x5007_0000	0x5007_FFFF	CPU sensor Group	-
0x5008_0000	0x5008_FFFF	Reserved	-
0x5009_0000	0x5009_FFFF	Reserved	-
0x500A_0000	0x500A_FFFF	GPU PIK	-
0x500B_0000	0x500B_FFFF	GPU sensor Group	-
0x500C_0000	0x500C_FFFF	NPU PIK (expansion)	-
0x500D_0000	0x500D_FFFF	NPU sensor Group (expansion)	-
0x500E_0000	0x500E_FFFF	Display PIK (expansion)	-
0x500F_0000	0x500F_FFFF	Display sensor Group (expansion)	-
0x5100_0000	0x517F_FFFF	System Access Port mapping on to Application Processor memory map region from 0x01_0000_0000 - 0x01_007F_FFFF	This section is for the SCP to access CLUS0-PPU<x> located inside the DSU through AP Cluster0 Utility space
0x5180_0000	0x5FFF_FFFF	Reserved	-
0x6000_0000	0x9FFF_FFFF	System Access Port mapping on to Application Processor memory map region from 0x00_4000_0000 to 0x00_7FFF_FFFF	-
0xA000_0000	0xDFFF_FFFF	System Access Port mapping on to Application Processor memory map region from 0x00_0000_0000 to 0x00_3FFF_FFFF	-
0xE000_0000	0xE000_0FFF	*Instrumentation Trace Macrocell* (ITM)	-
0xE000_1000	0xE000_1FFF	*Data Watchpoint Trace* (DWT)	-
0xE000_2000	0xE000_2FFF	FPB	-

Start address	End address	Region	Additional information
0xE000_3000	0xE000_DFFF	Reserved	-
0xE000_E000	0xE000_EFFF	SCS	-
0xE000_F000	0xE003_FFFF	Reserved	-
0xE004_0000	0xE004_0FFF	*Private Peripheral Bus* (PPB) (external) – TPIU	-
0xE004_1000	0xE004_1FFF	PPB (external) – *Embedded Trace Macrocell* (ETM)	-
0xE004_2000	0xE004_2FFF	PPB (external) – PIL CTI	-
0xE004_3000	0xE004_3FFF	Reserved	-
0xE004_4000	0xE004_4FFF	PPB (external) – ATB funnel	-
0xE004_5000	0xE004_5FFF	PPB (external) – ATB replicator	-
0xE004_6000	0xE004_6FFF	PPB (external) – CTI	-
0xE004_7000	0xE00D_FFFF	Reserved	-
0xE00F_E000	0xE00F_EFFF	PPB (external) – PIL expansion debug ROM	-
0xE00F_F000	0xE00F_FFFF	PPB (external) – debug ROM table	-
0xE010_0000	0xFFFF_FFFF	SCP AHB Expansion Port	-

7.2 Interrupt maps

The *System Control Processor* (SCP) and *Application Processor* (AP) have their own interrupt maps which are different from each other. These interrupt maps are detailed following.

7.2.1 AP interrupt map

This subsection describes the interrupt map for the *Application Processor* (AP).

The *Generic Interrupt Controller* (GIC) Architecture defines two different types of physical interrupts.

They are:

- *Private Peripheral Interrupts* (PPIs) that separately exist for every processor.
- *Shared Peripheral Interrupts* (SPIs) that are shared for all processors.

PPI and SPI interrupts have configurable options, including number of interrupts, Edge or Level triggered, and Polarity. For example, active-LOW, active-HIGH, or Rising edge.


Note

GIC PPI inputs are either active-LOW level sensitive, or triggered on a rising edge.

Table 7-3: AP interrupt map

Interrupt ID	Interrupt source	Description	Trigger	Polarity
0-15	SGI	Software Generated Interrupt	-	-
16	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
17	PMBIRQn	Statistical Profiling Extension interrupt (Only for ELP cores)	Level	Active-LOW
18	TRBIRQn	Statistical Profiling Extension interrupt (Only for ELP cores)	Level	Active-LOW
19	CNTHVSIRQn	Secure Virtual Timer event	Level	Active-LOW
20	CNTHPSIRQn	Secure Physical Timer event	Level	Active-LOW
21	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
22	COMMIRQn	Debug Communications Channel receive or transmit request	Level	Active-LOW
23	PMUIRQn	PMU interrupt	Level	Active-LOW
24	CTIIRQn	CTI interrupt	Edge	Rising edge
25	VCPUMNTIRQn	Virtual Maintenance Interrupt	Level	Active-LOW
26	CNTHPIRQn	Non-secure PL2 Timer event	Level	Active-LOW
27	CNTVIRQn	Virtual Timer event	Level	Active-LOW
28	CNTHVIRQn	EL2 virtual timer	Level	Active-LOW
29	CNTPSIRQn	Secure PL1 Physical Timer event	Level	Active-LOW
30	CNTPNSIRQn	Non-secure PL1 Physical Timer event	Level	Active-LOW
31-63	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
64	Cluster PMU	Cluster PMU Interrupt	Level	Active-LOW
65-69	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
70	CATUADDRERR	From the Debug block, indicate address translation error from CATU	Level	Active-HIGH
71	ETRBUFINT	From Embedded Trace Router (ETR)	Level	Active-HIGH
72	Crypto-TEE	From the crypto block, to be handled by Secure world SW	Level	Active-HIGH
73	Crypto-REE	From the crypto block	Level	Active-HIGH
74-82	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
83	STM-500 synchronization	-	Edge	Rising edge

Interrupt ID	Interrupt source	Description	Trigger	Polarity
84-85	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
86	Trusted Watchdog (WS0)	-	Level	Active-HIGH
87-90	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
91	AP_REFCLK Generic timer (Secure)	-	Level	Active-HIGH
92	AP_REFCLK Generic timer (Non-secure)	-	Level	Active-HIGH
93	Generic Watchdog WS0	-	Level	Active-HIGH
94	Generic Watchdog WS1	-	Level	Active-HIGH
95	AP_NS_UART_INT	-	Level	Active-HIGH
96	AP_S_UART_INT	-	Level	Active-HIGH
97	GPU interrupt	-	Level	Active-HIGH
98	GPU Job interrupt	-	Level	Active-HIGH
99	GPU MMU interrupt	-	Level	Active-HIGH
100	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
101	DPU IRQ0	-	Level	Active-HIGH
102	DPU IRQ1	-	Level	Active-HIGH
103	DPU TCU PMU interrupt	-	Edge	Rising edge
104	DPU TCU Event Queue Secure interrupt	-	Edge	Rising edge
105	DPU TCU CMD SYNC Secure interrupt	-	Edge	Rising edge
106	DPU TCU Global Secure interrupt	-	Edge	Rising edge
107	DPU TCU Event Queue Non-secure interrupt	-	Edge	Rising edge
108	DPU TCU CMD SYNC Non-secure interrupt	-	Edge	Rising edge
109	DPU TCU Global Non-secure interrupt	-	Edge	Rising edge
110-111	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
112	DPU TBU PMU interrupt, one per TBU	-	Edge	Rising edge
113-127	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-

Interrupt ID	Interrupt source	Description	Trigger	Polarity
128-255	Expansion interrupts	**IMPLEMENTATION DEFINED**, as RD-TC22 does not have expansion logic, these will be treated like Reserved interrupts and tied LOW. Blocks like the VPU, ISP, and Secure may have their interrupt connected to the Expansion interrupts.	-	-
256	System TCU PMU IRPT	-	Edge	Rising edge
257	System TCU Event Queue Secure IRPT	-	Edge	Rising edge
258	System TCU CMD SYNC Secure	-	Edge	Rising edge
259	System TCU Global Secure	-	Edge	Rising edge
260	System TCU Event Queue Non-secure	-	Edge	Rising edge
261	System TCU CMD SYNC Non-secure	-	Edge	Rising edge
262	System TCU Global Non-secure	-	Edge	Rising edge
263-283	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
284-288	System TBU PMU interrupt, one per TBU	-	Edge	Rising edge
289-347	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
348	Cluster0 SCP->AP MHU Secure	-	Level	Active-HIGH
349	Cluster0 SCP->AP MHU Non-secure, high-priority	-	Level	Active-HIGH
350	Cluster0 SCP->AP MHU Non-secure, low-priority	-	Level	Active-HIGH
351-475	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
476	NCI PMU MESHCLK	-	Level	Active-LOW
477	NCI PMU NOCSYSCLK	-	Level	Active-LOW
478	NCI PMU NOCMEMCLK	-	Level	Active-LOW
479	NCI PMU DPUCLK	-	Level	Active-LOW
480	NCI PMU GICCLK	-	Level	Active-LOW
481	NIC PMU GPUCLK	-	Level	Active-LOW
482	NCI PMU VPUCLK	-	Level	Active-LOW
483-491	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-

Interrupt ID	Interrupt source	Description	Trigger	Polarity
492	CI-700 PMU interrupt	-	Level	Active-HIGH
493	CI-700 Error Non-secure	-	Level	Active-HIGH
494	CI-700 Error Secure	-	Level	Active-HIGH
495	CI-700 Fault Non-secure	-	Level	Active-HIGH
496	CI-700 Fault Secure	-	Level	Active-HIGH
497	CI-700 MPAM Error Non-secure	-	Level	Active-HIGH
498	CI-700 MPAM Error Secure	-	Level	Active-HIGH
499-511	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
512	NPU0 JOB IRQ NS (expansion)	-	Edge	Rising edge
513	NPU0 DEBUG IRQ NS (expansion)	-	Edge	Rising edge
514	NPU0 ERR IRQ NS (expansion)	-	Level	Active-HIGH
515	NPU0 JOB IRQ S (expansion)	-	Edge	Rising edge
516	NPU0 DEBUG IRQ S (expansion)	-	Edge	Rising edge
517	NPU0 ERR IRQ S (expansion)	-	Level	Active-HIGH
518	NPU1 JOB IRQ NS (expansion)	-	Edge	Rising edge
519	NPU1 DEBUG IRQ NS (expansion)	-	Edge	Rising edge
520	NPU1 ERR IRQ NS (expansion)	-	Level	Active-HIGH
521	NPU1 JOB IRQ S (expansion)	-	Edge	Rising edge
522	NPU1 DEBUG IRQ S (expansion)	-	Edge	Rising edge
523	NPU1 ERR IRQ S (expansion)	-	Level	Active-HIGH
524	NPU2 JOB IRQ NS (expansion)	-	Edge	Rising edge
525	NPU2 DEBUG IRQ NS (expansion)	-	Edge	Rising edge
526	NPU2 ERR IRQ NS (expansion)	-	Level	Active-HIGH

Interrupt ID	Interrupt source	Description	Trigger	Polarity
527	NPU2 JOB IRQ S (expansion)	-	Edge	Rising edge
528	NPU2 DEBUG IRQ S (expansion)	-	Edge	Rising edge
529	NPU2 ERR IRQ S (expansion)	-	Level	Active-HIGH
530	NPU3 JOB IRQ NS (expansion)	-	Edge	Rising edge
531	NPU3 DEBUG IRQ NS (expansion)	-	Edge	Rising edge
532	NPU3 ERR IRQ NS (expansion)	-	Level	Active-HIGH
533	NPU3 JOB IRQ S (expansion)	-	Edge	Rising edge
534	NPU3 DEBUG IRQ S (expansion)	-	Edge	Rising edge
535	NPU3 ERR IRQ S (expansion)	-	Level	Active-HIGH

7.2.2 SCP interrupt map

This subsection describes the interrupt map for the *System Control Processor* (SCP).

The SCP receives interrupts from the following sources:

- Application processor system wakeup interrupts
- CoreSight power and reset request interrupts
- Internal SCP subsystem interrupts
- Expansion SCP interrupts

These interrupts are routed to the *Nested Vector Interrupt Controller* (NVIC) that is included in the Cortex-M3 processor, where they can be managed by software. The *Non-Maskable Interrupt* (NMI) and interrupt IDs 0-31 are wakeup sources.

The following table summarizes the regions of the SCP interrupt map.

Table 7-4: SCP interrupt map

Interrupt ID	Interrupt source	Description	Trigger	Polarity
NMI	SCP Watchdog	SCP Watchdog	Level	Active-HIGH
0	Reserved	-	-	-
1	CoreSight	CoreSight debug power up request	Edge	Rising edge

Interrupt ID	Interrupt source	Description	Trigger	Polarity
2	CoreSight	CoreSight system power up request	Edge	Rising edge
3	CoreSight	CoreSight debug reset request	Edge	Rising edge
4-7	Reserved	-	-	-
8	CPU	Cluster Fault/Error IRQ (combined MPAMNSIRQn, MPAMSIRQn CLUSTERCRITIRQn, CLUSTERERRIRQn, and CLUSTERFAULTIRQn)	Level	Active-HIGH
9	CPU	Core Fault/Error IRQ (combined COREERRIRQn, COMPLEXERRIRQn, COREFAULTIRQn, and COMPLEXFAULTIRQn)	Level	Active-HIGH
10-15	Reserved	-	-	-
16-31	External	SoC Expansion Wakeup Interrupt	-	-
32	SCP power control logic	Power control logic Interrupt	Level	Active-HIGH
33	SCP REFCLK Generic Timer	REFCLK Physical Timer interrupt	Level	Active-HIGH
34	MHU	*Message Handling Unit* (MHU) High-Priority Non-secure interrupt	Level	Active-HIGH
35	MHU	MHU Low Priority Non-secure interrupt	Level	Active-HIGH
36	MHU	MHU Secure interrupt	Level	Active-HIGH
37	CTI	*Cross Trigger Interface* (CTI) Trigger 0	Edge	Rising edge
38	CTI	CTI Trigger 1	Edge	Rising edge
39	GIC Error	*Generic Interrupt Controller* (GIC) FAULTINT Interrupt. SCP should treat this as a request to reset the whole SYSTOP.	Level	Active-HIGH
40	GIC Error	GIC ERRINT Interrupt. SCP should treat this as a request to reset the whole SYSTOP.	Level	Active-HIGH
41	DMC	Reserved	-	-
42	DMC	DMC0_combined_ecc_err_int	Level	Active-HIGH
43	DMC	DMC0_combined_misc_access_int	Level	Active-HIGH
44	DMC	TZC400-0 interrupt	Level	Active-HIGH
45	DMC	Reserved	-	-
46	DMC	DMC1_combined_ecc_err_int	Level	Active-HIGH
47	DMC	DMC1_combined_misc_access_int	Level	Active-HIGH
48	DMC	TZC400-1 interrupt	Level	Active-HIGH
49	DMC	Reserved	-	-

Interrupt ID	Interrupt source	Description	Trigger	Polarity
50	DMC	DMC2_combined_ecc_err_int	Reserved for C2	Active-HIGH
51	DMC	DMC2_combined_misc_access_int	Reserved for C2	Active-HIGH
52	DMC	TZC400-2 interrupt	Reserved for C2	Active-HIGH
53	DMC	Reserved	-	-
54	DMC	DMC3_combined_ecc_err_int	Reserved for C2	Active-HIGH
55	DMC	DMC3_combined_misc_access_int	Reserved for C2	Active-HIGH
56	DMC	TZC400-3 interrupt	Reserved for C2	Active-HIGH
57-72	Power Policy Units	CPU0 Core 0 to Core 15 PPU interrupt	Level	Active-HIGH
73	Power Policy Unit	CLUS0 PPU interrupt	Level	Active-HIGH
74	Power Policy Unit	SYS PPU0 interrupt	Level	Active-HIGH
75	Power Policy Unit	SYS PPU1 interrupt	Level	Active-HIGH
76	Power Policy Unit	GPU PPU interrupt	Level	Active-HIGH
77	Power Policy Unit	NPU PPU interrupt. Only implemented if this NPU exists in the reference subsystem	Level	Active-HIGH
78	Reserved	-	-	-
79	Power Policy Unit	DPU PPU1 interrupt	Level	Active-HIGH
80	Reserved	-	-	-
81	Power Policy Unit	Debug power control logic interrupt	Level	Active-HIGH
82-97	Reserved	-	-	-
98	PLL	Cluster0 *Phase-Locked Loop* (PLL) Lock	Edge	Rising edge
99	PLL	Reserved	Edge	Rising edge
100	PLL	GPU PLL Lock	Edge	Rising edge
101	Reserved	-	-	-
102	PLL	Sys PLL Lock	Edge	Rising edge
103	PLL	Display PLL Lock	Edge	Rising edge
104	PLL	CPU0 PLL0 Lock	Edge	Rising edge

Interrupt ID	Interrupt source	Description	Trigger	Polarity
105	PLL	CPU0 PLL1 Lock	Edge	Rising edge
106	PLL	CPU0 PLL2 Lock	Edge	Rising edge
107	PLL	CPU0 PLL3 Lock	Edge	Rising edge
108	PLL	CPU0 PLL4 Lock	Edge	Rising edge
109	PLL	CPU0 PLL5 Lock	Edge	Rising edge
110	PLL	CPU0 PLL6 Lock	Edge	Rising edge
111	PLL	CPU0 PLL7 Lock	Edge	Rising edge
112	PLL	CPU0 PLL8 Lock	Edge	Rising edge
113	PLL	CPU0 PLL9 Lock	Edge	Rising edge
114	PLL	CPU0 PLL10 Lock	Edge	Rising edge
115	PLL	CPU0 PLL11 Lock	Edge	Rising edge
116	PLL	CPU0 PLL12 Lock	Edge	Rising edge
117	PLL	CPU0 PLL13 Lock	Edge	Rising edge
118	PLL	CPU0 PLL14 Lock	Edge	Rising edge
119	PLL	CPU0 PLL15 Lock	Edge	Rising edge
120-127	Reserved	-	-	-
128-159	Expansion Interrupts	32 Interrupts for SCP Expansion	-	-

7.3 Register descriptions

This section contains information that can be used to program the subsystem. Each description provides details about a register, such as configurations, attributes, and bit assignments.

7.3.1 System ID registers

The *System ID* (SID) registers allow the embedding of identification and configuration information in a system.

Each component has the following identification registers:

- The Peripheral ID registers, PID0-PID4, provide information to the system about the component.
- The Component ID registers, CID0-CID3, specify the component type, or class, and identify Arm as the designer, along with other information.

The following table summarizes the registers and their corresponding address offsets for the SID. The SID_BASE is the base address of the SID, which is set to 0000_2A4A_0000. All registers support Secure and Non-secure accesses.

For more information on System ID register mapping, see AP memory map.

Table 7-5: System ID register summary

Offset	Name	Type	Reset	Width	Description
SID_BASE + 0x0040	SID_SYSTEM_ID	RO	0x00041712	32-bit	ID register
SID_BASE + 0x0050	SID_SOC_ID	RO	IMPLEMENTATION DEFINED	32-bit	ID register
SID_BASE + 0x0060	SID_NODE_ID	RO	System specific	32-bit	ID register
SID_BASE + 0x0070	SID_SYSTEM_CFG	RO	System specific	32-bit	Configuration register
SID_BASE + 0x0FD0	PID4	RO	0x00000004	32-bit	SID Peripheral ID 4 register
SID_BASE + 0x0FE0	PID0	RO	0x000000D2	32-bit	SID Peripheral ID 0 register
SID_BASE + 0x0FE4	PID1	RO	0x000000B0	32-bit	SID Peripheral ID 1 register
SID_BASE + 0x0FE8	PID2	RO	0x0000000B	32-bit	SID Peripheral ID 2 register
SID_BASE + 0x0FEC	PID3	RO	0x00000000	32-bit	SID Peripheral ID 3 register
SID_BASE + 0x0FF0	CID0	RO	0x0000000D	32-bit	SID Component ID 0 register
SID_BASE + 0x0FF4	CID1	RO	0x000000F0	32-bit	SID Component ID 1 register
SID_BASE + 0x0FF8	CID2	RO	0x00000005	32-bit	SID Component ID 2 register
SID_BASE + 0x0FFC	CID3	RO	0x000000B1	32-bit	SID Component ID 3 register

7.3.1.1 SID_SYSTEM_ID, System ID System Identification register

The SID_SYSTEM_ID register contains version information for the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0040`

Type

RO

Reset value

`0x00041712`

Bit descriptions

Table 7-6: SID_SYSTEM_ID bit descriptions

Bits	Name	Description
[31:28]	-	Reserved
[27:24]	MAJOR REVISION	Specifies the major revision number for the subsystem. Set to `0x0`.
[23:20]	MINOR REVISION	Specifies the major revision number for the subsystem. Set to `0x0`.
[19:12]	DESIGNER_ID	Specifies the identification code for the subsystem designer. Arm product with designer code `0x41`.
[11:0]	PART_NUMBER	Specifies the part number for the subsystem. Set to `0x712`, and used for Total Compute 2022 Reference Design (RD-TC22).

7.3.1.2 SID_SOC_ID register, System ID SoC Identification register

The SID_SOC_ID register contains version information for the *System on Chip* (SoC) that integrates the subsystem. The value of this register depends on the SID_SOC_ID signal.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0050`

Type

RO

Reset value

IMPLEMENTATION DEFINED

Bit descriptions

Table 7-7: SID_SOC_ID bit descriptions

Bits	Name	Description
[31:28]	-	Reserved
[27:24]	MAJOR REVISION	Specifies the major revision number for the subsystem. IMPLEMENTATION DEFINED
[23:20]	MINOR REVISION	Specifies the major revision number for the subsystem. IMPLEMENTATION DEFINED
[19:12]	DESIGNER_ID	Specifies the identification code for the subsystem designer. IMPLEMENTATION DEFINED
[11:0]	PART_NUMBER	Specifies the part number for the subsystem. IMPLEMENTATION DEFINED

7.3.1.3 SID_NODE_ID, System ID Node Identification register

The SID_CHIP_ID register contains information about the node when there are multiple sockets. The value of this register depends on the CHIPID and MULTI_CHIP_MODE signals.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + 0x0060

Type

RO

Reset value

System specific

Bit descriptions

Table 7-8: SID_NODE_ID bit descriptions

Bits	Name	Description
[31:8]	-	Reserved
8	MULTI_CHIP_MODE	Read-only register bits to indicate the multi-chip mode tie off value. • 0 – Single Chip • 1 – Multichip
[7:0]	NODE_NUMBER	Read-only register bits to indicate the NODE_Number or CHIP ID tie off value in the multi-chip mode. For single chip, this is 0.

7.3.1.4 SID_SYSTEM_CFG, System ID System Configuration register

The SID_SYSTEM_CFG register contains information about the subsystem configuration when multiple variants that use the same compute subsystem are supported. For example, when there is support for two DDR channel and four DDR channel variants. The value of this register depends on the SID_SYSTEM_CFG signal.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + 0x0070

Type

RO

Reset value

System specific

Bit descriptions

Table 7-9: SID_SYSTEM_CFG bit descriptions

Bits	Name	Description
[31:0]	CONFIG_NUMBER	System specific

7.3.1.5 PID4, System ID Peripheral ID 4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0FD0`

Type

RO

Reset value

`0x00000004`

Bit descriptions

Table 7-10: PID4 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).
[7:4]	SIZE	Indicates the log2 of the number of 4KB blocks occupied by the interface. Set to `0x0`.
[3:0]	DES_2	JEP106 Continuation Code identifies the designer. Set to `0x4` for Arm.

7.3.1.6 PID0, System ID Peripheral ID 0 register

The PID0 register contains the first eight bits of the identifier for the peripheral and remaining bits are reserved.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0FE0`

Type

RO

Reset value

`0x000000D2`

Bit descriptions

Table 7-11: PID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).
[7:0]	PART_0	Specifies bits [7:0] of the part identifier for the peripheral. Set to `0x44`.

7.3.1.7 PID1, System ID Peripheral ID 1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code, the second four bits of the identifier for the peripheral, and remaining bits are reserved.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0FE4`

Type

RO

Reset value

`0x000000B0`

Bit descriptions

Table 7-12: PID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).

Bits	Name	Description
[7:4]	DES_0	Specifies bits [3:0] of JEDEC JEP106 Identity code for the peripheral. Set to `0xB` for Arm.
[3:0]	PART_1	Specifies bits [11:8] of the part identifier for the peripheral. Set to `0x8`.

7.3.1.8 PID2, System ID Peripheral ID 2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0FE8`

Type

RO

Reset value

`0x0000000B`

Bit descriptions

Table 7-13: PID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).
[7:4]	REVISION	Specifies the major revision number for the block. Set to `0x0` for r0p0.
[3]	JEDEC	Specifies whether the JEDEC JEP106 identification scheme is in use. Set to `0x1`.
[2:0]	DES_1	Specifies bits [6:4] of the JEDEC JEP106 code for the peripheral. Set to `0x3` for Arm.

7.3.1.9 PID3, System ID Peripheral ID 3 register

The PID3 register is Reserved.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0FEC`

Type

RO

Reset value

`0x00000000`

Bit descriptions

Table 7-14: PID3 bit descriptions

Bits	Name	Description
[31:0]	-	Reserved, *Read-As-Zero* (RAZ).

7.3.1.10 CID0, System ID Component ID 0 register

The CID0 register contains the first eight bits that specify segment 0 of the preamble to the system ID component class identifier. The remaining bits are reserved.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0FF0`

Type

RO

Reset value

0x0000000D

Bit descriptions

Table 7-15: CID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).
[7:0]	COMP_ID0	Specifies segment 0 of the component to the code that identifies the system ID component class. Reads as 0x0D.

7.3.1.11 CID1, System ID Component ID 1 register

The CID1 register contains the first eight bits that specify segment 1 of the preamble to the system ID component class identifier. The remaining bits are reserved.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + 0x0FF4

Type

RO

Reset value

0x000000F0

Bit descriptions

Table 7-16: CID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).
[7:0]	COMP_ID1	Specifies segment 1 of the component to the code that identifies the system ID component class. Reads as 0xF0.

7.3.1.12 CID2, System ID Component ID 2 register

The CID2 register contains the first eight bits that specify segment 2 of the preamble to the system ID component class identifier. The remaining bits are reserved.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + `0x0FF8`

Type

RO

Reset value

`0x00000005`

Bit descriptions

Table 7-17: CID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).
[7:0]	COMP_ID2	Specifies segment 2 of the component to the code that identifies the system ID component class. Reads as `0x05`.

7.3.1.13 CID3, System ID Component ID 3 register

The CID3 register contains the first eight bits that specify segment 3 of the preamble to the system ID component class identifier. The remaining bits are reserved.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System ID registers

Address offset

SID_BASE + 0x0FFC

Type

RO

Reset value

0x000000B1

Bit descriptions

Table 7-18: CID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (RAZ).
[7:0]	COMP_ID3	Specifies segment 3 of the component to the code that identifies the system ID component class. Reads as 0xB1

7.3.2 Message Handling Unit registers

The MHU registers enable the configuration and operation of the Sender and Receiver.

The MHU implementation in Total Compute 2022 Reference Design (RD-TC22) only supports 32-bit word-aligned accesses. Unaligned accesses are treated as Read-As-Zero, Writes Ignored (RAZ/WI).

The MHU is made up of two memory-mapped register frames. The first frame is used by the Sender of the transfer, while the Receiver uses the second frame. The registers in each frame are listed in the following tables.

Table 7-19: MHU Sender frame register summary

Offset	Name	Type	Reset	Width	Description
0x000 – 0xF7C	Sender channel window registers	-	-	32-bit	MHU Sender channel window, see MHU channel window registers
0xF80	MHU_CFG	RO	See description	32-bit	Message Handling Unit Configuration The reset value for the MHU_CFG register is equal to the number of channels configured for the particular MHU. See MHU_CFG for more information.
0xF84	RESP_CFG	RW	0x00000000	32-bit	MHU Response Configuration
0xF88	ACCESS_REQUEST	RW	0x00000000	32-bit	MHU Access Request
0xF8C	ACCESS_READY	RO	0x00000000	32-bit	MHU Access Ready
0xF90	INT_ST	RO	0x00000000	32-bit	MHU Sender Interrupt Status
0xF94	INT_CLR	WO	0x00000000	32-bit	MHU Sender Interrupt Clear
0xF98	INT_EN	RW	0x00000000	32-bit	MHU Sender Interrupt Enable
0xF9C	-	-	-	-	Reserved

Offset	Name	Type	Reset	Width	Description
0xFA0	CHCOMB_INT_ST0	RO	0x00000000	32-bit	MHU Sender Channel combined interrupt status (0-31)
0xFA4	CHCOMB_INT_ST1	RO	0x00000000	32-bit	MHU Sender Channel combined interrupt status (32-63)
0xFA8	CHCOMB_INT_ST2	RO	0x00000000	32-bit	MHU Sender Channel combined interrupt status (64-95)
0xFAC	CHCOMB_INT_ST3	RO	0x00000000	32-bit	MHU Sender Channel combined interrupt status (96-123)
0xFB0 – 0xFC4	-	-	-	-	Reserved
0xFC8	IIDR	RO	0x0760043B	32-bit	MHU Implementer Identification register
0xFCC	AIDR	RO	0x00000011	32-bit	MHU Architecture Identification register
0x0FD0	PID4	RO	0x00000004	32-bit	MHU Peripheral ID 4
0x0FE0	PID0	RO	0x00000076	32-bit	MHU Peripheral ID 0
0x0FE4	PID1	RO	0x000000B0	32-bit	MHU Peripheral ID 1
0x0FE8	PID2	RO	0x0000000B	32-bit	MHU Peripheral ID 2
0x0FEC	PID3	RO	0x00000000	32-bit	MHU Peripheral ID 3
0x0FF0	COMPID0	RO	0x0000000D	32-bit	MHU Component ID 0 register
0x0FF4	COMPID1	RO	0x000000F0	32-bit	MHU Component ID 1 register
0x0FF8	COMPID2	RO	0x00000005	32-bit	MHU Component ID 2 register
0x0FFC	COMPID3	RO	0x000000B1	32-bit	MHU Component ID 3 register

Table 7-20: MHU Receiver frame register summary

Offset	Name	Type	Reset	Width	Description
0x000 – 0xF7C	Receiver channel window registers	-	-	-	MHU Receiver channel window, see MHU channel window registers
0xF80	MHU_CFG	RO	See description	32-bit	Message Handling Unit Configuration The reset value for the MHU_CFG register is equal to the number of channels configured for the particular MHU. See MHU_CFG for more information.
0xF84 – 0xF8C	-	RO	-	-	Reserved
0xF90	INT_ST	RO	0x00000000	32-bit	MHU Receiver Interrupt Status
0xF94	INT_CLR	WO	0x00000000	32-bit	MHU Receiver Interrupt Clear
0xF98	INT_EN	RW	0x00000000	32-bit	MHU Receiver Interrupt Enable
0xF9C	-	-	-	-	Reserved
0xFA0	CHCOMB_INT_ST0	RO	0x00000000	32-bit	MHU Receiver Channel combined interrupt status (0-31)
0xFA4	CHCOMB_INT_ST1	RO	0x00000000	32-bit	MHU Receiver Channel combined interrupt status (32-63)
0xFA8	CHCOMB_INT_ST2	RO	0x00000000	32-bit	MHU Receiver Channel combined interrupt status (64-95)
0xFAC	CHCOMB_INT_ST3	RO	0x00000000	32-bit	MHU Receiver Channel combined interrupt status (96-123)
0xFC8	IIDR	RO	0x0760043B	32-bit	MHU Implementer Identification register
0xFCC	AIDR	RO	0x00000011	32-bit	MHU Architecture Identification register
0x0FD0	PID4	RO	0x00000004	32-bit	MHU Peripheral ID 4

Offset	Name	Type	Reset	Width	Description
0x0FE0	PID0	RO	0x00000076	32-bit	MHU Peripheral ID 0
0x0FE4	PID1	RO	0x000000B0	32-bit	MHU Peripheral ID 1
0x0FE8	PID2	RO	0x0000000B	32-bit	MHU Peripheral ID 2
0x0FEC	PID3	RO	0x00000000	32-bit	MHU Peripheral ID 3
0x0FF0	COMPID0	RO	0x0000000D	32-bit	MHU Component ID 0 register
0x0FF4	COMPID1	RO	0x000000F0	32-bit	MHU Component ID 1 register
0x0FF8	COMPID2	RO	0x00000005	32-bit	MHU Component ID 2 register
0x0FFC	COMPID3	RO	0x000000B1	32-bit	MHU Component ID 3 register

7.3.2.1 MHU channel window registers

A channel window is a group of registers. The registers in the channel window vary between the Sender and Receiver frame views.

An MHU implementation can contain between 1 and 124 channels. The number of channels that are implemented can be discovered from the MHU_CFG.NUM_CH field. Each channel occupies eight 32-bit words in both the Sender and Receiver register maps. The address space that is allocated to channels that are not implemented is Reserved and treated as **RAZ/WI**.

The following tables list the registers in the Sender and Receiver channel windows.

Table 7-21: MHU Sender channel window register summary

Offset	Name	Type	Width	Description
0x00	CH_ST	RO	32-bit	MHU Channel Status
0x04 – 0x08	-	RO	-	Reserved
0x0C	CH_SET	WO	32-bit	MHU Channel Set
0x10 – 0x18	-	RO	-	Reserved

Table 7-22: MHU Receiver channel window register summary

Offset	Name	Type	Width	Description
0x00	CH_ST	RO	32-bit	MHU Channel Status
0x04	CH_ST_MSK	RO	32-bit	MHU Channel Status Masked
0x08	CH_CLR	WO	32-bit	MHU Channel Clear
0x0C	-	RO	-	Reserved
0x10	CH_MSK_ST	RO	32-bit	MHU Channel Mask Status
0x14	CH_MSK_SET	WO	32-bit	MHU Channel Mask Set
0x18	CH_MSK_CLR	WO	32-bit	MHU Channel Mask Clear
0x1C	-	RO	-	Reserved

7.3.2.1.1 CH_ST, MHU Channel Status register

The CH_ST register shows the state of the channel and is part of both the Receiver and Sender channel windows.

If the Receiver frame is reset, then the contents of the CH_ST register in the Sender frame are also reset. Software is responsible for handling any lost messages when the Receiver frame and CH_ST register are reset.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

MHU channel window registers

Address offset

`0x00`

Type

RO

Bit descriptions

Table 7-23: CH_ST bit descriptions

Bits	Name	Description	Type	Default
[31:0]	FLAGn, n = 0–31	Display the status of channel flags. Each bit can be used as an individual flag or bits can be grouped. The way in which the register is used depends on the transport protocol that is employed. Bits in this register are set by writing `0b1` to the corresponding bits in the CH_SET register. Writing `0b1` to bits in the CH_CLR register clears the corresponding bits in the CH_ST register. If software does one of the following, these corresponding action occurs: • If it sets a bit that is already set, the bit remains set. • If it clears a bit that is already cleared, the bit remains cleared. • If it sets and clears a bit at the same time, the bit remains set. Arm recommends that software follows the transport protocols that are defined for the MHU.	RO	`0x0000_0000`

7.3.2.1.2 CH_ST_MSK, MHU Channel Status Masked register

The CH_ST_MSK register shows the state of the channel with the channel mask applied, and is part of the Receiver channel window.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

MHU channel window registers

Address offset

0x04

Type

RO

Bit descriptions

Table 7-24: CH_ST_MSK bit descriptions

Bits	Name	Description	Type	Default
[31:0]	FLAG_MSKn, n = 0–31	Display the status of channel flags with the mask applied. When this register is nonzero, the interrupt for the channel is asserted. The value in this register is equal to CH_ST and CH_MSK_ST at the point at which the read occurs.	RO	0x0000_00000

7.3.2.1.3 CH_CLR, MHU Channel Clear register

The CH_CLR register resets bits in the Channel Status and Channel Status Masked registers, and is part of the Receiver channel window.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

MHU channel window registers

Address offset

0x08

Type

WO

Bit descriptions

Table 7-25: CH_CLR bit descriptions

Bits	Name	Description	Type	Default
[31:0]	FLAG_CLRn, n = 0–31	Clear the channel flags. Writing `0b1` to bits in this register clears the corresponding bits in the CH_ST and CH_ST_MSK registers. Writing `0b0` to bits in this register has no effect. Each bit always reads as `0b0`.	WO	`0x0000_0000`

7.3.2.1.4 CH_SET, MHU Channel Set register

The CH_SET register writes bits in the Channel Status and Channel Status Mask registers, and is part of the Sender channel window.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

MHU channel window registers

Address offset

`0x0C`

Type

WO

Bit descriptions

Table 7-26: CH_SET bit descriptions

Bits	Name	Description	Type	Default
[31:0]	FLAG_SETn, n = 0–31	Set the channel flags. Writing `0b1` to bits in this register sets the corresponding bits in the CH_ST register. Writing `0b0` to bits in this register has no effect. Each bit always reads as `0b0`.	WO	`0x0000_0000`

7.3.2.1.5 CH_MSK_ST, MHU Channel Mask Status register

The CH_MSK_ST register shows the state of the channel mask, and is part of the Receiver channel window. The channel mask is used with the Channel Status register to generate the channel status mask.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

MHU channel window registers

Address offset

`0x10`

Type

RO

Bit descriptions

Table 7-27: CH_MSK_ST bit descriptions

Bits	Name	Description	Type	Default
[31:0]	FLAG_MSKn, n = 0–31	Display the status of channel flag masks. A channel mask bit that is set to `0b0` indicates that the corresponding flag bit is unmasked. When a bit is unmasked, the equivalent bits in the CH_ST and CH_ST_MSK registers have the same value. A channel mask bit that is set to `0b1` indicates that the corresponding flag bit is masked. When a bit is masked, the equivalent bit in the CH_ST_MSK register always reads as `0b0`.	RO	`0x0000_0000`

7.3.2.1.6 CH_MSK_SET, MHU Channel Mask Set register

The CH_MSK_SET register writes bits in the channel mask and is part of the Receiver channel window.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

MHU channel window registers

Address offset

`0x14`

Type

WO

Bit descriptions

Table 7-28: CH_MSK_SET bit descriptions

Bits	Name	Description	Type	Default
[31:0]	FLAG_MSK_SETn, n = 0–31	Set the channel flag masks. Writing `0b1` to bits in this register sets the corresponding bits in the CH_MSK_ST register. Writing `0b0` to bits in this register has no effect. Each bit always reads as `0b0`.	WO	`0x0000_0000`

7.3.2.1.7 CH_MSK_CLR, MHU Channel Mask Clear register

The CH_MSK_CLR register resets bits in the channel mask and is part of the Receiver channel window.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

MHU channel window registers

Address offset

`0x18`

Type

WO

Bit descriptions

Table 7-29: CH_MSK_CLR bit descriptions

Bits	Name	Description	Type	Default
[31:0]	FLAG_MSK_CLRn, n = 0–31	Clear the channel flag masks. Writing `0b1` to bits in this register clears the corresponding bits in the CH_MSK_ST register. Writing `0b0` to bits in this register has no effect. Each bit always reads as `0b0`.	WO	`0x0000_0000`

7.3.2.2 MHU_CFG, Message Handling Unit Configuration register

The MHU_CFG register shows the number of channels that are implemented in the MHU.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

0xF80

Type

RO

Bit descriptions

Table 7-30: MHU_CFG bit descriptions

Bits	Name	Description	Type	Default
[31:7]	-	Reserved.	RO	0x0000_0000
[6:0]	NUM_CH	Specifies the number of MHU channels that are implemented. The value of the field indicates the number of channels, up to a maximum of 124 (0x7C). The values 0x00, 0x7D, 0x7E, and 0x7F are reserved.	RO	CFG_DEF

7.3.2.3 RESP_CFG, MHU Response Configuration register

The RESP_CFG register shows the number of channels that are implemented in the MHU.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

0xF84

Type

RW

Bit descriptions

Table 7-31: RESP_CFG bit descriptions

Bits	Name	Description	Type	Default
[31:1]	-	Reserved.	RO	0x0000_0000
[0]	NR_RESP	Specifies the response that is generated when the Sender attempts to access any channel window register while the ACCESS_READY.ACC_RDY field is set to 0b0. This setting indicates that the Receiver is not in a state in which it can accept a transfer. When the Receiver is not ready, channel window register access attempts by the Sender are treated as **RAZ/WI**. With the RESP_CFG.NR_RESP field set to 0b0, an error is not generated. If this field is set to 0b1, then an error is generated.	RO	CFG_DEF

7.3.2.4 ACCESS_REQUEST, MHU Access Request register

The ACCESS_REQUEST register is used by the Sender to require that the Receiver enters a state in which the Receiver can accept a transfer.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xF88`

Type

RW

Bit descriptions

Table 7-32: ACCESS_REQUEST bit descriptions

Bits	Name	Description	Type	Default
[31:1]	-	Reserved.	RO	`0x0000_0000`
[0]	ACC_REQ	Requests that the Receiver prepares to accept a transfer. A setting of `0b0` for this field indicates that the Receiver does not need to prepare for a transfer. If this field is set to `0b1`, then the Receiver is requested to prepare to accept a transfer.	RW	`0b0`

7.3.2.5 ACCESS_READY, MHU Access Ready register

The ACCESS_READY register shows whether the Receiver is in a state in which it can accept a transfer from the Sender.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xF8C`

Type

RO

Bit descriptions

Table 7-33: ACCESS_READY bit descriptions

Bits	Name	Description	Type	Default
[31:1]	-	Reserved.	RO	0x0000_0000
[0]	ACC_RDY	Specifies whether the Receiver is able to accept a transfer. A setting of 0b0 for this field indicates that the Receiver is not able to accept a transfer. If this field is set to 0b1, then the Receiver is able to accept a transfer.	RW	0b0

7.3.2.6 INT_ST, MHU Sender Interrupt Status register

The INT_ST register shows whether the Ready to Not Ready and Not Ready to Ready interrupts have been generated.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

0xF90

Type

RO

Bit descriptions

Table 7-34: INT_ST bit descriptions

Bits	Name	Description	Type	Default
[31:3]	-	Reserved.	RO	0x0000_0000
[2]	CHCOMB	Channel combined interrupt status. **0b0** No interrupt has occurred on any Channel. **0b1** An interrupt has occurred on at least one Channel. There is no corresponding bit in the INT_CLR register. To clear this interrupt, software must clear the underlying interrupt.	RO	0b0

Bits	Name	Description	Type	Default
[1]	R2NR	Ready to not ready interrupt status. **0b0** Ready to not ready interrupt has not occurred. **0b1** Ready to not ready interrupt has occurred.	RO	0b0
[0]	NR2R	Not ready to ready interrupt status. **0b0** Not ready to ready interrupt has not occurred. **0b1** Not ready to ready interrupt has occurred.	RO	0b0

7.3.2.7 INT_CLR, MHU Sender Interrupt Clear register

The INT_CLR register resets the ready to not ready and not ready to ready interrupts.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

0xF94

Type

WO

Bit descriptions

Table 7-35: INT_CLR bit descriptions

Bits	Name	Description	Type	Default
[31:2]	-	Reserved.	RO	0x0000_0000
[1]	R2NR	Clears the ready to not ready interrupt. • Writing 0b1 to this field clears the ready to not ready interrupt. • Writing 0b0 to this field has no effect.	WO	0b0
[0]	NR2R	Clears the not ready to ready interrupt. • Writing 0b1 to this field clears the not ready to ready interrupt. • Writing 0b0 to this field has no effect.	WO	0b0

7.3.2.8 INT_EN, MHU Sender Interrupt Enable register

The INT_EN register activates and deactivates generation of the ready to not ready and not ready to ready interrupts.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xF98`

Type

RW

Bit descriptions

Table 7-36: INT_EN bit descriptions

Bits	Name	Description	Type	Default
[31:3]	-	Reserved.	RO	`0x0000_0000`
[2]	CHCOMB	Channel combined interrupt enable. **0b0** Combined interrupt is disabled. **0b1** Combined interrupt is enabled.	RO	`0b1`
[1]	R2NR	Ready to not ready interrupt enable. **0b0** Ready to not ready interrupt has not occurred. **0b1** Ready to not ready interrupt has occurred.	RW	`0b0`
[0]	NR2R	Not ready to ready interrupt enable. **0b0** Not ready to ready interrupt has not occurred. **0b1** Not ready to ready interrupt has occurred.	RW	`0b0`

7.3.2.9 INT_ST, MHU Receiver Interrupt Status register

The INT_ST register shows whether the ready to not ready and not ready to ready interrupts have been generated.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xF90`

Type

RO

Bit descriptions

Table 7-37: INT_ST bit descriptions

Bits	Name	Description	Type	Default
[31:3]	-	Reserved.	RO	`0x0000_0000`
[2]	CHCOMB	Channel combined interrupt status. **`0b0`** No interrupt has occurred on any Channel. **`0b1`** An interrupt has occurred on at least one Channel. There is no corresponding bit in the INT_CLR register. To clear this interrupt, software must clear the underlying interrupt.	RO	`0b0`
[1:0]	-	Reserved.	RO	`0x0000_0000`

7.3.2.10 INT_CLR, MHU Receiver Interrupt Clear register

This register is included for completeness and for an orthogonal set of registers between Sender and Receiver. It has no functionality in the receiver.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

0xF94

Type

WO

Bit descriptions

Table 7-38: INT_CLR bit descriptions

Bits	Name	Description	Type	Default
[31:0]	-	Reserved	RO	0x0000_0000

7.3.2.11 INT_EN, MHU Receiver Interrupt Enable register

The INT_EN register activates and deactivates generation of the ready to not ready and not ready to ready interrupts.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

0xF98

Type

RW

Bit descriptions

Table 7-39: INT_EN bit descriptions

Bits	Name	Description	Type	Default
[31:3]	-	Reserved.	RO	0x0000_0000
[2]	CHCOMB	Channel combined interrupt enable. **0b0** Combined interrupt is disabled. **0b1** Combined interrupt is enabled.	RO	0b1
[1:0]	-	Reserved.	RO	0x0000_0000

7.3.2.12 CHCOMB_INT_ST<0-3>, MHU Sender Channel Combined Interrupt Status 0-3 register

The CHCOMB_INT_ST register indicates the channel interrupt status for the respective channels as shown in the bit descriptions table below.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xFA0 – 0xFAC`

Type

RO

Bit descriptions

Table 7-40: CHCOMB_INT_ST<0-3> bit descriptions

Bits	Name	Description	Type	Default
[31:0]	CHCOMB_INT_ST{x}	Channel interrupt status. Each bit indicates whether a Channel has a pending interrupt or not. • CHCOMB_INT_ST0 has the status for Channels 0 to 31, starting with Channel 0 at bit 0. • CHCOMB_INT_ST1 has the status for Channels 32 to 63, starting with Channel 32 at bit 0. • CHCOMB_INT_ST2 has the status for Channels 64 to 95, starting with Channel 64 at bit 0. • CHCOMB_INT_ST3 has the status for Channels 96 to 123, starting with Channel 96 at bit 0. • A bit relating to an unimplemented Channel is Reserved and treated as **RAZ/WI**. • CHCOMB_INT_ST3 bits [31:28] are always Reserved and treated as **RAZ/WI**.	RO	`0x0000_0000`

7.3.2.13 CHCOMB_INT_ST<0-3>, MHU Receiver Channel Combined Interrupt Status 0-3 register

The CHCOMB_INT_ST register indicates the channel interrupt status for the respective channels as shown in the table below.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xFA0 – 0xFAC`

Type

RO

Bit descriptions

Table 7-41: CHCOMB_INT_ST<0-3> bit descriptions

Bits	Name	Description	Type	Default
[31:0]	CHCOMB_INT_ST{x}	Channel interrupt status. Each bit indicates whether a Channel has a pending interrupt or not. • CHCOMB_INT_ST0 has the status for Channels 0 to 31, starting with Channel 0 at bit 0. • CHCOMB_INT_ST1 has the status for Channels 32 to 63, starting with Channel 32 at bit 0. • CHCOMB_INT_ST2 has the status for Channels 64 to 95, starting with Channel 64 at bit 0. • CHCOMB_INT_ST3 has the status for Channels 96 to 123, starting with Channel 96 at bit 0. • A bit relating to an unimplemented Channel is reserved and treated as **RAZ/WI**. • CHCOMB_INT_ST3 bits [31:28] are always reserved and treated as **RAZ/WI**.	RO	`0x0000_0000`

7.3.2.14 IIDR, MHU Implementer Identification register

The IIDR contains information about the MHU implementation.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xFC8`

Type

RO

Reset value

`0x0760043B`

Bit descriptions

Table 7-42: IIDR bit descriptions

Bits	Name	Description	Type	Default
[31:20]	PRODUCT_ID	Specifies the MHU part identifier.	RO	`0x076`
[19:16]	VARIANT	Specifies the MHU major revision number.	RO	`0x0`
[15:12]	REVISION	Specifies the MHU minor revision number.	RO	`0x0`
[11:0]	IMPLEMENTER	Specifies the JEDEC JEP106 manufacturers identification code for the MHU implementer. • Bits[11:8] contain the JEP106 continuation code for the implementer. • Bit[7] must always be 0. • Bits[6:0] give the JEP106 identity code for the implementer.	RO	`0x43B`

7.3.2.15 AIDR, MHU Architecture Identification register

The AIDR contains the MHU architecture version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit register

Address offset

`0xFCC`

Type

RO

Bit descriptions

Table 7-43: AIDR bit descriptions

Bits	Name	Description	Type	Default
[31:8]	-	Reserved.	RO	0x00_0000
[7:4]	ARCH_MAJOR_REV	Specifies the MHU major architecture revision number. A value of 0x1 indicates that the MHU conforms to MHU architecture version 2. The setting 0x0 is Reserved. When the ARCH_MAJOR_REV field is set to 0x0, the values in the ARCH_MINOR_REV field and IIDR register are **RAZ**. Software must determine in a platform-specific manner the MHU architecture version to which the component conforms.	RO	0x1
[3:0]	ARCH_MINOR_REV	Specifies the MHU minor architecture revision number. A value of 0x0 indicates that the architecture minor revision number is 0. All other values are reserved.	RO	0x0

7.3.2.16 PID4, MHU Peripheral ID 4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

0x0FD0

Type

RO

Reset value

0x00000004

Bit descriptions

Table 7-44: PID4 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	RESERVED	Reserved. **RAZ**.	RO	0x0
[7:4]	SIZE	Specifies the number of 4KB address blocks that are required to access the registers, expressed in powers of 2. Set to 0x0.	RO	0x0
[3:0]	DES_2	Specifies the JEDEC JEP106 continuation code for the peripheral, which indicates the number of 0x7F continuation characters in the manufacturer identity code.	RO	0x4

7.3.2.17 PID0, MHU Peripheral ID 0 register

The PID0 register contains the first eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

`0x0FE0`

Type

RO

Reset value

`0x00000076`

Bit descriptions

Table 7-45: PID0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	RESERVED	Reserved. **RAZ**.	RO	`0x0`
[7:0]	PART_0	Specifies bits[7:0] of the part identifier for the peripheral.	RO	`0x76`

7.3.2.18 PID1, MHU Peripheral ID 1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code and the second eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

`0x0FE4`

Type

RO

Reset value

0x000000B0

Bit descriptions

Table 7-46: PID1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	RESERVED	Reserved. **RAZ**.	RO	0x0
[7:4]	DES_0	Specifies bits[3:0] of the JEDEC JEP106 identity code for the peripheral.	RO	0xB
[3:0]	PART_1	Specifies bits[11:8] of the part identifier for the peripheral.	RO	0x0

7.3.2.19 PID2, MHU Peripheral ID 2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

0x0FE8

Type

RO

Reset value

0x0000000B

Bit descriptions

Table 7-47: PID2 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	RESERVED	Reserved. **RAZ**.	RO	0x0
[7:4]	REVISION	Specifies the major revision number for the block.	RO	0x0
[3]	JEDEC	Specifies whether the JEDEC JEP106 identification scheme is in use.	RO	0b1
[2:0]	DES_1	Specifies bits[6:4] of the JEDEC JEP106 designer code for the peripheral.	RO	0b011

7.3.2.20 PID3, MHU Peripheral ID 3 register

The PID3 register provides information about any modifications to the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

`0x0FEC`

Type

RO

Reset value

`0x00000000`

Bit descriptions

Table 7-48: PID3 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	RESERVED	Reserved. **RAZ**.	RO	`0x0`
[7:4]	REVAND	Manufacturer revision number: This field indicates minor errata fixes specific to this design, for example metal fixes after implementation	RO	`0x0`
[7:0]	CMOD	Cusomter modification number: incremented on authorized customer modifications	RO	`0x0`

7.3.2.21 COMPID0, MHU Component ID 0 register

The COMPID0 register contains segment 0 of the preamble to the MHU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

0x0FF0

Type

RO

Reset value

0x0000000D

Bit descriptions

Table 7-49: COMPID0 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved. **RAZ/WI**.	RO	-
[7:0]	PRMBL_0	Specifies segment 0 of the preamble to the code that identifies the MHU component class.	RO	0x0D

7.3.2.22 COMPID1, MHU Component ID 1 register

The COMPID1 register contains segment 1 of the preamble and the MHU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

0x0FF4

Type

RO

Reset value

0x000000F0

Bit descriptions

Table 7-50: COMPID1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	RAZ	Reserved	RO	-
[7:4]	CLASS	Specifies a code that identifies the MHU component class	RO	-
[3:0]	PRMBL_1	Specifies segment 1 of the preamble to the code that identifies the MHU component class	RO	0xF0

7.3.2.23 COMPID2, MHU Component ID 2 register

The COMPID2 register contains segment 2 of the preamble to the MHU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

`0x0FF8`

Type

RO

Reset value

`0x00000005`

Bit descriptions

Table 7-51: COMPID2 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved. **RAZ/WI**.	RO	-
[7:0]	PRMBL_2	Specifies segment 2 of the preamble to the code that identifies the MHU component class.	RO	`0x05`

7.3.2.24 COMPID3, MHU Component ID 3 register

The COMPID3 register contains segment 3 of the preamble to the MHU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Message Handling Unit registers

Address offset

`0x0FFC`

Type

RO

Reset value

0x000000B1

Bit descriptions

Table 7-52: COMPID3 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved. **RAZ/WI**.	RO	-
[7:0]	PRMBL_3	Specifies segment 3 of the preamble to the code that identifies the MHU component class.	RO	0xB1

7.3.3 REFCLK CNTControl

The REFCLK counter is an implementation of the memory mapped counter defined by the *Arm® Architecture Reference Manual for A-profile architecture*.

It implements both the standard REFCLK CNTControl frame and the REFCLK CNTRead frame in the AP memory map. Their base address and accessibility is also defined in this map.

This counter implements three extra registers in the CNTControl frame that are not defined in the *Arm® Architecture Reference Manual for A-profile architecture*. The following table lists these extra registers in REFCLK CNTControl frame and their offset addresses.

Table 7-53: Extra registers in REFCLK CNTControl frame

Offset	Name	Type	Reset	Width	Description
0xC0	CNTSCR	R/W	0X00	32-bit	Control Counter Synchronization Control register
0xC4	CNTSVL	RO	0x00	32-bit	Control Counter Synchronized Counter Lower Value register
0xC8	CNTSVU	RO	0x00	32-bit	Control Counter Synchronized Counter Upper Value register

7.3.3.1 CNTSCR, Control Counter Synchronization Control register

The CNTSCR register controls how the enabling and disabling of the REFCLK generic counter is performed.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

REFCLK CNTControl frame registers

Address offset

0xC0

Type

RW

Reset value

0x00

Bit descriptions

Table 7-54: CNTSCR bit descriptions

Bits	Type	Default	Name	Description
[31:1]	RW	0x0	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[0]	RW	0x0	ENSYNC	Controls the way the Counter Control register EN bit operates: 0 – The counter is enabled or disabled immediately. 1 – The enabling of the counter is delayed until just after the next rising edge at the REFCLK. Disabling the counter is not delayed.

7.3.3.2 CNTSVL, Control Synchronized Counter Value Lower Value register

The CNTSVL register reads the value of the counter sampled on the rising edge of REFCLK. This register returns the lower word CNTSV[31:0].

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

REFCLK CNTControl frame registers

Address offset

0xC4

Type

RO

Reset value

0x00

Bit descriptions

Table 7-55: CNTSVL bit descriptions

Bits	Type	Default	Name	Description
[31:0]	RW	0x0	CNTSVL	REFCLK-sampled value of the counter, lower word CNTSV[31:0]

7.3.3.3 CNTSVU, Control Synchronized Counter Value Upper Value register

The CNTSVU register reads the value of the counter sampled on the rising edge of REFCLK. This register returns the upper word CNTSV[63:32].

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

REFCLK CNTControl frame registers

Address offset

0xC8

Type

RO

Reset value

0x00

Bit descriptions

Table 7-56: CNTSUV bit descriptions

Bits	Type	Default	Name	Description
[31:0]	RW	0x0	CNTSVU	REFCLK-sampled value of the counter, upper word CNTSV[63:32]

7.3.4 Generic timer registers

The generic timer registers provide access to and control of the timer.

The Total Compute 2022 Reference Design (RD-TC22) supports various timer frames.

The timer base addresses are mapped in the AP memory map and SCP memory map:

- Pn_REFCLK per AP Generic wake-up Timers for n=0 to NUM_CLUSTERS. These are mapped in the AP and SCP memory maps.
- SCP_REFCLK Generic Timer is mapped in the SCP memory map.
- Two additional REFCLK generic timers, one Secure and one Non-secure, are solely for general-purpose use by the AP.

The following table lists the generic timer register summary. Only the PIDn and CIDn registers are described in this section. For more information on the other registers, see the *Arm® Architecture Reference Manual for A-profile architecture*.

Table 7-57: Generic timer register summary

Offset	Name	Type	Reset	Width	Description
0x00	CNTPCT[31:0]	RO	-	32-bit	Physical Count register
0x04	CNTPCT[63:32]	RO	-	32-bit	Physical Count register
0x08	CNTVCT[31:0]	RO	-	32-bit	Virtual Count register
0x0C	CNTVCT[63:32]	RO	-	32-bit	Virtual Count register
0x10	CNTFRQ	RW	-	32-bit	Counter Frequency register
0x14	CNTPL0ACR	RW	0x0	32-bit	Counter PL0 Access Control register
0x18	CNTVOFF[31:0]	RO	-	32-bit	Virtual Offset register
0x1C	CNTVOFF[63:32]	RO	-	32-bit	Virtual Offset register
0x20	CNTP_CVAL[31:0]	RW	0x0	32-bit	Physical Timer CompareValue register
0x24	CNTP_CVAL[63:32]	RW	0x0	32-bit	Physical Timer CompareValue register
0x28	CNTP_TVAL	RW	0x0	32-bit	Physical Timer Value register
0x2C	CNTP_CTL	RW	0x0	32-bit	Physical Timer Control register
0x30	CNTV_CVAL[31:0]	RW	0x0	32-bit	Virtual Timer Compare Value register
0x34	CNTV_CVAL[63:32]	RW	0x0	32-bit	Virtual Timer Compare Value register
0x38	CNTV_TVAL	RW	0x0	32-bit	Virtual Timer Value register
0x3C	CNTV_CTL	RW	0x0	32-bit	Virtual Timer Control register
0x40 – 0xFCC	-	-	-	32-bit	Reserved, UNK/SBZP
0xFD0	PID4	RO	0x04	32-bit	Generic Timer Peripheral Identification 4 register
0xFD4 – 0xFDC	-	-	-	-	Reserved, UNK/SBZP
0xFE0	PID0	RO	[e]	32-bit	Generic Timer Peripheral Identification 0 register
0xFE4	PID1	RO	0xB0	32-bit	Generic Timer Peripheral Identification 1 register
0xFE8	PID2	RO	0x0B	32-bit	Generic Timer Peripheral Identification 2 register
0xFEC	PID3	RO	0x00	32-bit	Generic Timer Peripheral Identification 3 register
0xFF0	CID0	RO	0x0D	32-bit	Generic Timer Component Identification 0 register
0xFF4	CID1	RO	0xF0	32-bit	Generic Timer Component Identification 1 register
0xFF8	CID2	RO	0x05	32-bit	Generic Timer Component Identification 2 register
0xFFC	CID3	RO	0xB1	32-bit	Generic Timer Component Identification 3 register

7.3.4.1 PID4, Generic Timer Peripheral ID 4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

0xFD0

Type

RO

Reset value

0x04

Bit descriptions

Table 7-58: PID4 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[7:4]	SIZE	Specifies the number of 4KB address blocks that are required to access the registers, expressed in powers of 2.
[3:0]	DES_2	Specifies the JEDEC JEP106 continuation code for the peripheral.

7.3.4.2 PID0, Generic Timer Peripheral ID 0 register

The PID0 register contains the first eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

0xFE0

Type

RO

Bit descriptions

When read from the CNTBASEN APB interface

Table 7-59: PID0 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[7:0]	PART_0	Specifies bits[7:0] of the part identifier for the peripheral.

When read from the CNTPL0BASEN APB interface:

Table 7-60: PID0 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:0]	PART_0	Specifies bits[7:0] of the part identifier for the peripheral.

7.3.4.3 PID1, Generic Timer Peripheral ID 1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code and the second four bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

`0xFE4`

Type

RO

Reset value

`0xB0`

Bit descriptions

Table 7-61: PID1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved, *Read-As-Zero*, *Writes Ignored* (RAZ/WI).	RO	`0x0`
[7:4]	DES_0	Specifies bits[3:0] of the JEDEC JEP106 identity code for the peripheral.	RO	`0xB`
[3:0]	PART_1	Specifies bits[11:8] of the part identifier for the peripheral.	RO	`0x0`

7.3.4.4 PID2, Generic Timer Peripheral ID 2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

`0xFE8`

Type

RO

Reset value

`0x0B`

Bit descriptions

Table 7-62: PID2 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[7:4]	REVISION*	Specifies the major revision number of the block. Set to `0x0` for r0p0. Note: For ECO purposes REVISION bits are driven by active register, which is loaded at reset. This ensures that the logic cone for this register is preserved.
[3]	JEDEC	Specifies whether the JEDEC JEP106 identification scheme is in use. Set to `0x1`.
[2:0]	DES_1	Specifies bits[6:4] of the JEDEC JEP106 designer code for the peripheral.

7.3.4.5 PID3, Generic Timer Peripheral ID 3 register

The PID3 register contains the manufacturer revision number and a value that is incremented with each authorized customer modification.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

`0xFEC`

Type

RO

Reset value

0x00

Bit descriptions

Table 7-63: PID3 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[7:4]	REVAND	Specifies the manufacturer revision number.
[3:0]	CMOD	Specifies the authorized customer modification increment.

7.3.4.6 CID0, Generic Timer Component ID 0 register

The CID0 register contains segment 0 of the preamble to the generic timer component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

0xFF0

Type

RO

Reset value

0x0D

Bit descriptions

Table 7-64: CID0 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[7:0]	PRMBL_0	Specifies segment 0 of the preamble to the code that identifies the counter-timer component class.

7.3.4.7 CID1, Generic Timer Component ID 1 register

The CID1 register contains segment 1 of the preamble and the generic timer component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-65: CID1 bit descriptions

Bits	Name	Description
[31:8]	Reserved, RAZ	Reserved, *Read-As-Zero* (RAZ).
[7:4]	CLASS	Specifies a code that identifies the generic timer component class.
[3:0]	PRMBL_1	Specifies segment 1 of the preamble to the code that identifies the generic timer component class.

7.3.4.8 CID2, Generic Timer Component ID 2 register

The CID2 register contains segment 2 of the preamble to the generic timer component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

0xFF8

Type

RO

Reset value

0x05

Bit descriptions

Table 7-66: CID2 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[7:0]	PRMBL_2	Specifies segment 2 of the preamble to the code that identifies the counter-timer component class.

7.3.4.9 CID3, Generic Timer Component ID 3 register

The CID3 register contains segment 3 of the preamble to the generic timer component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Generic timer registers

Address offset

0xFFC

Type

RO

Reset value

0xB1

Bit descriptions

Table 7-67: CID3 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[7:0]	PRMBL_3	Specifies segment 3 of the preamble to the code that identifies the counter-timer component class.

7.3.5 PPU power control registers

The *Power Policy Unit* (PPU) power control registers are defined inside *Arm® Power Policy Unit Architecture Specification, version 1.1*.

The PPUs used in the design include:

- CLUS0-PPU<0~x> where x is the number of processor cores
- SYS-PPU0
- SYS-PPU1
- DBG-PPU
- GPU-PPU0
- SCP-PPU

7.3.6 Core Manager and Power Control (CPU PIK) registers

The core manager and power control registers enable configuration of reset values and clock settings for subsystems.

The following table lists the fully configured power control register summary.

Table 7-68: Core Manager and Power Control register summary

Offset	Name	Type	Reset	Width	Description
0x0000	CLUSTER_CONFIG	RW	0x0	32-bit	Static Config for Global features
0x0004 – 0x000C	-	RO	-	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
0x0100 – 0x01FC	PE_STATIC_CONFIG	RW	0x0	32-bit	Processing Element (PE) static configuration
0x0200 – 0x07FC	-	RO	-	-	Reserved, RAZ/WI
0x0700	DBGCLK_CLKDIV	RW	0x0000001F	32-bit	DEBUG Clock Divider Control register
0x0704	DBGCLK_CLKSEL	RW	0x00000001	32-bit	DEBUG Clock Select Control register
0X0708 – 0x0800	-	RW	-	-	Reserved, RAZ/WI
0x0804 – 0x080C	-	RO	-	-	Reserved, RAZ/WI
0x0810	CLUSTER_PCLK_CLKDIV	RW	0x0000001F	32-bit	PCLK Clock Divider Control register
0x0814 – 0x081C	-	RO	-	-	Reserved, RAZ/WI
0x0820	CLUSTER_ATCLK_CLKDIV	RW	0x0000001F	32-bit	ATCLK Clock Divider Control register
0x0824 – 0x082C	-	RO	-	-	Reserved, RAZ/WI
0x0830	CLUSTER_GICCLK_CLKDIV	RW	0x0000001F	32-bit	DBGCLK Divider Control register
0x0834 – 0x083C	-	RO	-	-	Reserved, RAZ/WI

Offset	Name	Type	Reset	Width	Description
0x0840	-	RW	-	-	Reserved, RAZ/WI
0x0844 – 0x084C	-	RO	-	-	Reserved, RAZ/WI
0x0850	SCLK_CLKDIV	RW	0x0000001F	32-bit	SCLK Divider Control register
0x0854	-	RO	-	-	Reserved, RAZ/WI
0x0858 – 0x08FC	-	RO	-	-	Reserved, RAZ/WI
0x0900 – 0x09FC	CORE<x>CLK configuration registers	RW	-	-	Core Clock Divider and Control registers <x> = 0, 1, 2 ... number of cores - 1 The registers are assigned to cores contiguously with higher indices remaining unused
0x0A00 – 0x0A7C	COMPLEX<x>CLK configuration registers	RW	-	-	Complex Core Clock Divider and Control registers <x> = 0, 1, 2 ... number of core complexes - 1 The registers are assigned to complexes contiguously with higher indices remaining unused
0x0A80 – 0x0BFC	-	RO	-	-	Reserved, RAZ/WI
0x0C00	CLKFORCE_STATUS_CLUSTER	RO	0x0	32-bit	Cluster Clock Force Statuse register
0x0C04	CLKFORCE_SET_CLUSTER	WO	-	32-bit	Cluster Clock Force Set register
0x0C08	CLKFORCE_CLR_CLUSTER	WO	-	32-bit	Cluster Clock Force Clear register
0x0C10	CLKFORCE_STATUS_CORE	RO	0x0	32-bit	Core Clock Force Statuse register
0x0C14	CLKFORCE_SET_CORE	WO	-	32-bit	Core Clock Force Set register
0x0C18	CLKFORCE_CLR_CORE	WO	-	32-bit	Core Clock Force Clear register
0x0C20	CLKFORCE_STATUS_COMPLEX	RO	0x0	32-bit	Complex Clock Force Status register
0x0C24	CLKFORCE_SET_COMPLEX	WO	-	32-bit	Complex Clock Force Set register
0x0C28	CLKFORCE_CLR_COMPLEX	WO	-	32-bit	Complex Clock Force Clear register
0x0C2C – 0x0DFC	-	RO	-	-	Reserved, RAZ/WI
0x0E00	ERRIRQ_STATUS_CLUSTER	RO	0x0	32-bit	Cluster Error Recovery Interrupt Status register
0x0E04	ERRIRQ_STATUS_CORE	RO	0x0	32-bit	Core Error Recovery Interrupt Status register
0x0E08	ERRIRQ_STATUS_COMPLEX	RO	0x0	32-bit	Complex Error Recovery Interrupt Status register
0x0E10	FAULTIRQ_STATUS_CLUSTER	RO	0x0	32-bit	Cluster Fault Handling Interrupt Status register
0x0E14	FAULTIRQ_STATUS_CORE	RO	0x0	32-bit	Core Fault Handling Interrupt Status register
0x0E18	FAULTIRQ_STATUS_COMPLEX	RO	0x0	32-bit	Complex Fault Handling Interrupt Status register
0x0E1C – 0x0FB0	-	RO	-	-	Reserved, RAZ/WI
0x0FB0	CAP3	RO	0x0000_0007	32-bit	Capability Definition register
0x0FB4	CAP2	RO	0x0	32-bit	Capability Definition register
0x0FB8	CAP1	RO	0x70FC_001F	32-bit	Capability Definition register
0x0FBC	CAP0	RO	0x70FC_001F	32-bit	Capability Definition Register

Offset	Name	Type	Reset	Width	Description
0x0FC0	PWR_CTRL_CONFIG	RO	0x0014_0000	32-bit	Reserved
0x0FC4 – 0x0FCC	-	RO	-	-	Reserved, RAZ/WI
0x0FD0	PID4	RO	0x0000_0044	32-bit	Core Manager and Power Control Peripheral ID 4 register
0x0FD4	PID5	RO	0x0	-	Core Manager and Power Control Peripheral ID 5 register
0x0FD8	PID6	RO	0x0	-	Core Manager and Power Control Peripheral ID 6 register
0x0FDC	PID7	RO	0x0	-	Core Manager and Power Control Peripheral ID 7 register
0x0FE0	PID0	RO	0x0000_00B8	32-bit	Core Manager and Power Control Peripheral ID 8 register
0x0FE4	PID1	RO	0x0000_00B0	32-bit	Core Manager and Power Control Peripheral ID 1 register
0x0FE8	PID2	RO	0x0000_0007	32-bit	Core Manager and Power Control Peripheral ID 2 register
0x0FEC	PID3	RO	0x0	32-bit	Core Manager and Power Control Peripheral ID 3 register
0x0FF0	CID0	RO	0x0000_000D	32-bit	Core Manager and Power Control Component ID 0 register
0x0FF4	CID1	RO	0x0000_00F0	32-bit	Core Manager and Power Control Component ID 1 register
0x0FF8	CID2	RO	0x0000_0005	32-bit	Core Manager and Power Control Component ID 2 register
0x0FFC	CID3	RO	0x0000_00B1	32-bit	Core Manager and Power Control Component ID 0 register

7.3.6.1 CLUSTER_CONFIG, Cluster Static Configuration register

This register sets cluster static configurations. The value should be set before releasing cluster reset.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0000

Type

RW

Reset value

0x0

Bit descriptions

Table 7-69: CLUSTER_CONFIG bit descriptions

Bits	Name	Description
[31:2]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[1]	ELADISABLE	Disables Embedded Logic Analyzer hardware inside the DSU debug block. This pin is only sampled during reset of the cluster debug logic.
[0]	CRYPTODISABLE	Disables Cryptographic Extensions. This pin is only sampled during reset of the processor.

7.3.6.2 PE_STATIC_CONFIG, Processing Element Static Configuration region

The PE Config region is split into one section per each *Processing Element* (PE) implemented within the cluster.

Each section is made up of four registers that are available at the following offset:

0x0

The PE<x> static configuration register is available. For bit descriptions, see PE<x> Static Configuration register bit descriptions.

0x4

The Processing Element Reset Vector Base Address Lower Word register, RVBARADDR<x>_LOW, is available. For bit descriptions, see RVBARADDR<x>_LOW register bit descriptions.

0x8

The Processing Element Reset Vector Base Address Upper Word register, RVBARADDR<x>_UP, is available at `0x8`. For bit descriptions, see RVBARADDR<x>_UP bit descriptions.

0xC

[31:0] bits are Reserved (RAZ/WI).

Configurations

This region is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0100 – 0x01FC`

Type

RW

Reset value

0x0

Bit descriptions

The following table summarizes the bit descriptions of the PE<x> Static Configuration register.

Table 7-70: PE<x> Static Configuration register bit descriptions

Bits	Name	Description
[31:12]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[11:8]	PDPSTATE	If the core supports PDPSTATE, this drives the PDPSTATE input. If unsupported, treated as **RAZ/WI**. Sampled during reset.
[7]	DISPBLK	If the core supports DISPBLK, this drives the DISPBLK input. If unsupported, treated as **RAZ/WI**. Sampled during reset.
[6:5]	MPMMSTATE	If MPMM is enabled, selects which settings to use (core defined). If unsupported, treated as **RAZ/WI**. Sampled during reset.
[4]	MPMMEN	Enable for max power mitigation control, for cores that support that functionality. If unsupported, treated as **RAZ/WI**. Sampled during reset.
[3:1]	-	Reserved, RAZ/WI
[0]	CFGEND	Endianness of processor. Sampled during reset.

The following table summarizes the bit descriptions of the Processing Element Reset Vector Base Address Lower Word register.

Table 7-71: RVBARADDR<x>_LOW register bit descriptions

Bits	Name	Description
[31:2]	RVBARADDR	Holds bits [31:2] of the RVBARADDR<x> input to the cluster/PE. Sampled during reset.
[1:0]	Reserved	Reserved, **RAZ/WI**.

The following table summarizes the bit descriptions of the Processing Element Reset Vector Base Address Upper Word register.

Table 7-72: RVBARADDR<x>_UP bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, **RAZ/WI**.
[7:0]	RVBARADDR	Holds bits [39:32] of the RVBARADDR<x> input to the cluster/PE. Sampled during reset.

7.3.6.3 DBGCLK_CLKDIV, Debug Clock Divider Control register

Describes the Debug Clock Divider Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0700`

Type

RW

Reset value

`0x0000001F`

Bit descriptions

Table 7-73: DBGCLK_CLKDIV bit descriptions

Bits	Name	Description
[31:24]	ENTRYDELAY	Number of clock cycles between the clock not being required and the request to dynamically gate it.
[23:21]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[20:16]	CLKDIV_CUR	Holds the currently selected divide ratio for the clock.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKDIV	New clock divide ratio value. Divider value is CLKDIV +1, so writing 0 gives a divide ratio of 1.

7.3.6.4 DBGCLK_CLKSEL, Debug Clock Select Control register

This register lets the user read the current source clock or change the source clock of DBGCLK, the cluster debug clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0704`

Type

RW

Reset value

`0x00000001`

Bit descriptions

Table 7-74: DBGCLK_CLKSEL bit descriptions

Bits	Name	Description
[31:21]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[20:16]	CLKSEL_CUR	Holds the currently selected clock source. Mapping matches that of the CLKSEL field.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKSEL	Select new clock source. `0x0` is clock gated, `0x1` is REFCLK, `0x2` is SYSPLLCLK, all other values are Reserved and writing them is **UNPREDICTABLE**.

7.3.6.5 CLUSTER_PCLK_CLKDIV, Clock divider control register

This register lets the user read the current source clock or change the source clock of PCLK, the cluster peripheral clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0810`

Type

RW

Reset value

`0x0000001F`

Bit descriptions

Table 7-75: CLUSTER_PCLK_CLKDIV bit descriptions

Bits	Name	Description
[31:24]	ENTRYDELAY	Number of clock cycles between the clock not being required and the request to dynamically gate it.
[23:21]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[20:16]	CLKDIV_CUR	Holds the currently selected divide ratio for the clock. For SCLK this field is Reserved.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKDIV	New clock divide ratio value. Divider value is CLKDIV +1, so writing 0 gives a divide ratio of 1. For SCLK this field is Reserved.

7.3.6.6 CLUSTER_ATCLK_CLKDIV, Clock divider control register

This register lets the user read the current source clock or change the source clock of ATCLK, the cluster core trace clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0820`

Type

RW

Reset value

`0x0000001F`

Bit descriptions

Table 7-76: CLUSTER_ATCLK_CLKDIV bit descriptions

Bits	Name	Description
[31:24]	ENTRYDELAY	Number of clock cycles between the clock not being required and the request to dynamically gate it.
[23:21]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).
[20:16]	CLKDIV_CUR	Holds the currently selected divide ratio for the clock. For SCLK this field is Reserved.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKDIV	New clock divide ratio value. Divider value is CLKDIV +1, so writing 0 gives a divide ratio of 1. For SCLK this field is Reserved.

7.3.6.7 CLUSTER_GICCLK_CLKDIV, Clock divider control registers

This register lets the user read the current source clock or change the source clock of GICCLK, the clock for cluster interrupt controllers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0830

Type

RW

Reset value

0x0000001F

Bit descriptions

Table 7-77: CLUSTER_GICCLK_CLKDIV bit descriptions

Bits	Name	Description
[31:24]	ENTRYDELAY	Number of clock cycles between the clock not being required and the request to dynamically gate it.
[23:21]	-	Reserved, **RAZ/WI**.
[20:16]	CLKDIV_CUR	Holds the currently selected divide ratio for the clock. For SCLK this field is Reserved.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKDIV	New clock divide ratio value. Divider value is CLKDIV +1, so writing 0 gives a divide ratio of 1. For SCLK this field is Reserved.

7.3.6.8 SCLK_CLKDIV, Clock divider control register

This register lets the user read the current source clock or change the source clock of SCLK, the processor cluster clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0850

Type

RW

Reset value

0x0000001F

Bit descriptions

Table 7-78: SCLK_CLKDIV bit descriptions

Bits	Name	Description
[31:24]	ENTRYDELAY	Number of clock cycles between the clock not being required and the request to dynamically gate it.
[23:21]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[20:16]	CLKDIV_CUR	Holds the currently selected divide ratio for the clock. For SCLK this field is Reserved.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKDIV	New clock divide ratio value. Divider value is CLKDIV +1, so writing 0 gives a divide ratio of 1. For SCLK this field is Reserved.

7.3.6.9 CORE<x>CLK and COMPLEX<x>CLK Configuration registers

These registers let the user read the current status of the clock controls or update the controls of CORE<x>CLK and COMPLEX<x>CLK, the processor core and complex clocks. The controls include source clock, clock divider values, and clock modulation values.

Each core clock and complex clock has an associated set of registers as follows:

- CORE<x>CLK, Core Clock Divider and Control registers
- COMPLEX<x>CLK, Complex Clock Divider and Control registers

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

CORE<x>CLK `0x0900 – 0x097C`

COMPLEX<x>CLK `0x0A00 – 0x0A7C`

Type

RW

Bit descriptions

Table 7-79: CORE<x>CLK and COMPLEX<x>CLK Configuration registers bit descriptions

Offset	Name	Description
`0x0`	<clock name>_CLKDIV	Divider controls.
`0x4`	<clock name>_CLKSEL	Selector controls.
`0x8`	<clock name>_CLKMOD	Modulator controls.
`0xC`	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI).

The following table lists the CLKDIV register bit assignment.

Table 7-80: CLKDIV bit descriptions

Bits	Name	Description
[31:24]	ENTRYDELAY	Number of clock cycles between the clock not being required and the request to dynamically gate it.
[23:21]	-	Reserved, RAZ/WI
[20:16]	CLKDIV_CUR	Holds the currently selected divide ratio for the clock.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKDIV	New clock divide ratio value. Divider value is CLKDIV +1, so writing 0 gives a divide ratio of 1.

The following table lists the CLKSEL register bit assignment.

Table 7-81: CLKSEL bit descriptions

Bits	Name	Description
[31:21]	-	Reserved, RAZ/WI
[20:16]	CLKSEL_CUR	Holds the currently selected clock source. Mapping matches that of the CLKSEL field.
[15:5]	-	Reserved, RAZ/WI
[4:0]	CLKSEL	Select new clock source. • `0x0` is clock gated • `0x1` is REFCLK • `0x2` is CPUPLLCLK0 • `0x4` is CPUPLLCLK1 • `0x8` is CPUPLLCLK2 • `0x10` is CPUPLLCLK3 • All other values are Reserved and writing them as **UNPREDICTABLE**.

The following table lists the CLKMOD, Clock Modulator register bit assignment.

Table 7-82: CLKMOD bit descriptions

Bits	Name	Description
[31:24]	NUMERATOR_CUR	Current numerator value
[23:16]	DENOMINATOR_CUR	Current denominator value
[15:8]	NUMERATOR	Clock modulator numerator. Writing 0 disables the clock on all cycles.
[7:0]	DENOMINATOR	Clock modulator denominator. A value of 0 is Reserved and results in **UNPREDICTABLE** behavior. If DENOMINATOR is less than NUMERATOR, then the clock is enabled on all cycles.

7.3.6.10 CLKFORCE_STATUS_CLUSTER, Cluster Clock Force Status register

The Clock Force Status register records whether the dynamic clock gating associated with that clock is enabled (0) or disabled (1).

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C00

Type

RO

Reset value

0x0

Bit descriptions

Table 7-83: CLKFORCE_STATUS_CLUSTER bit descriptions

Bits	Name	Description
[31:7]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[6]	PERIPHCLKFORCE	Clock force for PERIPHCLK
[5]	SCLKFORCE	Clock force for SCLK
[4]	DBGCLKFORCE	Clock force for DBGCLK
[3]	CLUSTER_GICCLKFORCE	Clock force for CLUSTER_GICCLK
[2]	CLUSTER_ATCLKFORCE	Clock force for CLUSTER_ATCLK
[1]	CLUSTER_PCLKFORCE	Clock force for CLUSTER_PCLK
[0]	-	Reserved, RAZ/WI

7.3.6.11 CLKFORCE_SET_CLUSTER, Cluster Clock Force Set register

Writing 1 to a bit in this register will disable dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C04

Type

WO

Reset value

0x0

Bit descriptions

Table 7-84: CLKFORCE_SET_CLUSTER bit descriptions

Bits	Name	Description
[31:7]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[6]	PERIPHCLKFORCE	Clock force for PERIPHCLK
[5]	SCLKFORCE	Clock force for SCLK
[4]	DBGCLKFORCE	Clock force for DBGCLK
[3]	CLUSTER_GICCLKFORCE	Clock force for CLUSTER_GICCLK
[2]	CLUSTER_ATCLKFORCE	Clock force for CLUSTER_ATCLK
[1]	CLUSTER_PCLKFORCE	Clock force for CLUSTER_PCLK
[0]	-	Reserved, RAZ/WI

7.3.6.12 CLKFORCE_CLR_CLUSTER, Cluster Clock Force Clear register

Writing 1 to a bit in this register will enable dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C08

Type

WO

Reset value

0x0

Bit descriptions

Table 7-85: CLKFORCE_CLR_CLUSTER bit descriptions

Bits	Name	Description
[31:7]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[6]	PERIPHCLKFORCE	Clock force for PERIPHCLK
[5]	SCLKFORCE	Clock force for SCLK
[4]	DBGCLKFORCE	Clock force for DBGCLK
[3]	CLUSTER_GICCLKFORCE	Clock force for CLUSTER_GICCLK
[2]	CLUSTER_ATCLKFORCE	Clock force for CLUSTER_ATCLK
[1]	CLUSTER_PCLKFORCE	Clock force for CLUSTER_PCLK
[0]	-	Reserved, RAZ/WI

7.3.6.13 CLKFORCE_STATUS_CORE, Core Clock Force Status register

The Clock Force Status register records whether the dynamic clock gating associated with that clock is enabled (0) or disabled (1).

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C10

Type

RO

Reset value

0x0

Bit descriptions

Table 7-86: CLKFORCE_STATUS_CORE bit descriptions

Bits	Name	Description
[31:24]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[23:8]	CORE<x>CLKFORCE	One bit per core clock. Any unused bits are treated as Reserved (RAZ/WI)
[7:0]	-	Reserved, RAZ/WI

7.3.6.14 CLKFORCE_SET_CORE, Core Clock Force Set register

Writing 1 to a bit in this register will disable dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C14

Type

WO

Reset value

0x0

Bit descriptions

Table 7-87: CLKFORCE_SET_CORE bit descriptions

Bits	Name	Description
[31:24]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[23:8]	CORE<x>CLKFORCE	One bit per core clock. Any unused bits are treated as Reserved (RAZ/WI)
[7:0]	-	Reserved, RAZ/WI

7.3.6.15 CLKFORCE_CLR_CORE, Core Clock Force Clear register

Writing 1 to a bit in this register will enable dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C18

Type

WO

Reset value

0x0

Bit descriptions

Table 7-88: CLKFORCE_CLR_CORE bit descriptions

Bits	Name	Description
[31:24]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[23:8]	CORE<x>CLKFORCE	One bit per core clock. Any unused bits are treated as Reserved (RAZ/WI)
[7:0]	-	Reserved, RAZ/WI

7.3.6.16 CLKFORCE_STATUS_COMPLEX, Complex Clock Force Status register

The Clock Force Status register records whether the dynamic clock gating associated with that clock is enabled (0) or disabled (1).

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C20

Type

RO

Reset value

0x0

Bit descriptions

Table 7-89: CLKFORCE_STATUS_COMPLEX bit descriptions

Bits	Name	Description
[31:24]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[23:8]	COMPLEX<x>CLKFORCE	One bit per complex clock. Any unused bits are treated as Reserved (RAZ/WI)
[7:0]	-	Reserved, RAZ/WI

7.3.6.17 CLKFORCE_SET_COMPLEX, Complex Clock Force Set register

Writing 1 to a bit in this register will disable dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C24

Type

WO

Reset value

0x0

Bit descriptions

Table 7-90: CLKFORCE_SET_COMPLEX bit descriptions

Bits	Name	Description
[31:24]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[23:8]	COMPLEX<x>CLKFORCE	One bit per complex clock. Any unused bits are treated as Reserved (RAZ/WI)
[7:0]	-	Reserved, RAZ/WI

7.3.6.18 CLKFORCE_CLR_COMPLEX, Complex Clock Force Clear register

Writing 1 to a bit in this register will enable dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0C28

Type

WO

Reset value

0x0

Bit descriptions

Table 7-91: CLKFORCE_CLR_COMPLEX bit descriptions

Bits	Name	Description
[31:24]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
[23:8]	COMPLEX<x>CLKFORCE	One bit per complex clock. Any unused bits are treated as Reserved (RAZ/WI)
[7:0]	-	Reserved, RAZ/WI

7.3.6.19 ERRIRQ_STATUS_CLUSTER register

The status of *Reliability, Availability, and Serviceability* (RAS) error recovery interrupt signal of the cluster. For more information, see the cluster document.

For all valid ERRIRQn bits, a value of 0x0 indicates the interrupt has been asserted, 0x1 indicates the interrupt has not been asserted.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0E00

Type

RO

Bit descriptions

Table 7-92: ERRIRQ_STATUS_CLUSTER bit descriptions

Bits	Name	Description
[31:20]	-	Reserved, Read-As-Zero (RAZ)
[19]	MPAMNSIRQn	MPAMNSIRQ status
[18]	MPAMSIRQn	MPAMSIRQ status
[17]	CLUSTERCRITIRQn	CLUSTERCRITIRQ status
[16:1]	-	Reserved (RAZ)

Bits	Name	Description
[0]	CLUSTERERRIRQn	Cluster ERRIRQn status.

7.3.6.20 ERRIRQ_STATUS_CORE register

The status of the RAS error recovery interrupt signal of the processor core. For more information, see the processor core document.

For all valid ERRIRQn bits, a value of `0x0` indicates the interrupt has been asserted, `0x1` indicates the interrupt has not been asserted.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0E04`

Type

RO

Bit descriptions

Table 7-93: ERRIRQ_STATUS_CORE bit descriptions

Bits	Name	Description
[31:16]	-	Reserved, Read-As-Zero (RAZ)
[15:0]	COREERRIRQn	1 bit per core. If a core is not implemented, the associated bit is Reserved, RAZ

7.3.6.21 ERRIRQ_STATUS_COMPLEX register

The status of the RAS error recovery interrupt signal of the processor complex. For more information, see the processor complex document.

For all valid ERRIRQn bits, a value of `0x0` indicates the interrupt has been asserted, `0x1` indicates the interrupt has not been asserted.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0E08`

Type

RO

Bit descriptions

Table 7-94: ERRIRQ_STATUS_COMPLEX bit descriptions

Bits	Name	Description
[31:16]	-	Reserved, Read-As-Zero (RAZ)
[15:0]	COMPLEXERRIRQn	1 bit per complex – if a core is not implemented, the associated bit is Reserved, RAZ

7.3.6.22 FAULTIRQ_STATUS_CLUSTER register

The status of the RAS fault handling interrupt signal of the cluster. For more information, see the cluster document.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0E10`

Type

RO

Bit descriptions

For all valid FAULTIRQn bits, a value of `0x0` indicates the interrupt has been asserted, `0x1` indicates the interrupt has not been asserted.

Table 7-95: FAULTIRQ_STATUS_CLUSTER register bit descriptions

Bits	Name	Description
[31:1]	-	Reserved, *Read-As-Zero* (RAZ).
[0]	CLUSTERFAULTIRQn	Cluster FAULTIRQn status.

7.3.6.23 FAULTIRQ_STATUS_CORE register

The status of the RAS fault handling interrupt signal of the processor core. For more information, see the processor core document.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0E14`

Type

RO

Bit descriptions

For all valid FAULTIRQn bits, a value of `0x0` indicates the interrupt has been asserted, `0x1` indicates the interrupt has not been asserted.

Table 7-96: FAULTIRQ_STATUS_CORE register bit descriptions

Bits	Name	Description
[31:1]	-	Reserved, *Read-As-Zero* (RAZ).
[15:0]	COREFAULTIRQn	1 bit per core – if a core is not implemented, the associated bit is Reserved, RAZ

7.3.6.24 FAULTIRQ_STATUS_COMPLEX register

The status of the RAS fault handling interrupt signal of the processor complex. For more information, see the processor complex document.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0E18`

Type

RO

Bit descriptions

For all valid FAULTIRQn bits, a value of `0x0` indicates the interrupt has been asserted, `0x1` indicates the interrupt has not been asserted.

Table 7-97: FAULTIRQ_STATUS_COMPLEX register bit descriptions

Bits	Name	Description
[31:1]	-	Reserved, *Read-As-Zero* (RAZ).
[15:0]	COMPLEXFAULTIRQn	1 bit per core – if a core is not implemented, the associated bit is Reserved, RAZ

7.3.6.25 CAP3, Core Manager and Power Control Capability Definition 3 register

This register indicates various capabilities of the cluster. For the type of capabilities indicated by the register, see the Bit descriptions section of the register definition.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FB0`

Type

RO

Reset value

`0x0000_0007`

Bit descriptions

Table 7-98: CAP3 bit descriptions

Bits	Name	Description
[31:16]	-	Reserved
[15]	CPUPLLCLK15_NOT_PRESENT	Indicates whether CPUPLLCLK15 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[14]	CPUPLLCLK14_NOT_PRESENT	Indicates whether CPUPLLCLK14 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[13]	CPUPLLCLK13_NOT_PRESENT	Indicates whether CPUPLLCLK13 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[12]	CPUPLLCLK12_NOT_PRESENT	Indicates whether CPUPLLCLK12 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[11]	CPUPLLCLK11_NOT_PRESENT	Indicates whether CPUPLLCLK11 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[10]	CPUPLLCLK10_NOT_PRESENT	Indicates whether CPUPLLCLK10 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[9]	CPUPLLCLK9_NOT_PRESENT	Indicates whether CPUPLLCLK9 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[8]	CPUPLLCLK8_NOT_PRESENT	Indicates whether CPUPLLCLK8 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[7]	CPUPLLCLK7_NOT_PRESENT	Indicates whether CPUPLLCLK7 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[6]	CPUPLLCLK6_NOT_PRESENT	Indicates whether CPUPLLCLK6 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[5]	CPUPLLCLK5_NOT_PRESENT	Indicates whether CPUPLLCLK5 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[4]	CPUPLLCLK4_NOT_PRESENT	Indicates whether CPUPLLCLK4 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present

Bits	Name	Description
[3]	CPUPLLCLK3_NOT_PRESENT	Indicates whether CPUPLLCLK3 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[2]	CPUPLLCLK2_NOT_PRESENT	Indicates whether CPUPLLCLK2 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[1]	CPUPLLCLK1_NOT_PRESENT	Indicates whether CPUPLLCLK1 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present
[0]	CPUPLLCLK0_NOT_PRESENT	Indicates whether CPUPLLCLK0 has been implemented or not. • `0x0` – PLL input present • `0x1` – PLL input not present

7.3.6.26 CAP2, Core Manager and Power Control Capability Definition 2 register

This register indicates various capabilities of the cluster. For the type of capabilities indicated by the register, see the Bit descriptions section of the register definition.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FB4`

Type

RO

Reset value

`0x0`

Bit descriptions

All THREADS_CORE<x> fields have the following mapping: `0x0` – 1 thread, `0x1` – 2 threads, all other values are Reserved.

Table 7-99: CAP2 bit descriptions

Bits	Name	Description
[31:30]	THREADS_CORE15	Number of threads in core 15

Bits	Name	Description
[29:28]	THREADS_CORE14	Number of threads in core 14
[27:26]	THREADS_CORE13	Number of threads in core 13
[25:24]	THREADS_CORE12	Number of threads in core 12
[23:22]	THREADS_CORE11	Number of threads in core 11
[21:20]	THREADS_CORE10	Number of threads in core 10
[19:18]	THREADS_CORE9	Number of threads in core 9
[17:16]	THREADS_CORE8	Number of threads in core 8
[15:14]	THREADS_CORE7	Number of threads in core 7
[13:12]	THREADS_CORE6	Number of threads in core 6
[11:10]	THREADS_CORE5	Number of threads in core 5
[9:8]	THREADS_CORE4	Number of threads in core 4
[7:6]	THREADS_CORE3	Number of threads in core 3
[5:4]	THREADS_CORE2	Number of threads in core 2
[3:2]	THREADS_CORE1	Number of threads in core 1
[1:0]	THREADS_CORE0	Number of threads in core 0

7.3.6.27 CAP1, Core Power Control Capability Definition 1 register

This register indicates various capabilities of the cluster. For the type of capabilities indicated by the register, see the Bit descriptions section of the register definition.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FB8`

Type

RO

Reset value

`0x70FC_001F`

Bit descriptions

Table 7-100: CAP1 bit descriptions

Bits	Name	Description
[31:16]	-	Reserved

Bits	Name	Description
[15]	CORE15CLK_NOT_PRESENT	Indicates whether CORE15CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[14]	CORE14CLK_NOT_PRESENT	Indicates whether CORE14CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[13]	CORE13CLK_NOT_PRESENT	Indicates whether CORE13CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[12]	CORE12CLK_NOT_PRESENT	Indicates whether CORE12CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[11]	CORE11CLK_NOT_PRESENT	Indicates whether CORE11CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[10]	CORE10CLK_NOT_PRESENT	Indicates whether CORE10CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[9]	CORE9CLK_NOT_PRESENT	Indicates whether CORE9CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[8]	CORE8CLK_NOT_PRESENT	Indicates whether CORE8CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[7]	CORE7CLK_NOT_PRESENT	Indicates whether CORE7CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[6]	CORE6CLK_NOT_PRESENT	Indicates whether CORE6CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[5]	CORE5CLK_NOT_PRESENT	Indicates whether CORE5CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[4]	CORE4CLK_NOT_PRESENT	Indicates whether CORE4CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[3]	CORE3CLK_NOT_PRESENT	Indicates whether CORE3CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present

Bits	Name	Description
[2]	CORE2CLK_NOT_PRESENT	Indicates whether CORE2CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[1]	CORE1CLK_NOT_PRESENT	Indicates whether CORE1CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[0]	CORE0CLK_NOT_PRESENT	Indicates whether CORE0CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present

7.3.6.28 CAP0, Core Manager and Power Control Capability Definition 0 register

This register indicates various capabilities of the cluster. For the type of capabilities indicated by the register, see the Bit descriptions section of the register definition.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FBC`

Type

RO

Reset value

`0x70FC_001F`

Bit descriptions

Table 7-101: CAP0 bit descriptions

Bits	Name	Description
[31:28]	NUM_PE	Number of processing elements in the cluster. `0x0` – 1 PE up to `0xF` – 16 PEs
[27]	CLUSSYNC	Indicates whether the Cluster is synchronous to the interconnect or not. • `0x0` – asynchronous • `0x1` – synchronous (includes N:1 sync)
[26:16]	-	Reserved

Bits	Name	Description
[15]	COMPLEX15CLK_NOT_PRESENT	Indicates whether COMPLEX15CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[14]	COMPLEX14CLK_NOT_PRESENT	Indicates whether COMPLEX14CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[13]	COMPLEX13CLK_NOT_PRESENT	Indicates whether COMPLEX13CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[12]	COMPLEX12CLK_NOT_PRESENT	Indicates whether COMPLEX12CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[11]	COMPLEX11CLK_NOT_PRESENT	Indicates whether COMPLEX11CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[10]	COMPLEX10CLK_NOT_PRESENT	Indicates whether COMPLEX10CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[9]	COMPLEX9CLK_NOT_PRESENT	Indicates whether COMPLEX9CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[8]	COMPLEX8CLK_NOT_PRESENT	Indicates whether COMPLEX8CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[7]	COMPLEX7CLK_NOT_PRESENT	Indicates whether COMPLEX7CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[6]	COMPLEX6CLK_NOT_PRESENT	Indicates whether COMPLEX6CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[5]	COMPLEX5CLK_NOT_PRESENT	Indicates whether COMPLEX5CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[4]	COMPLEX4CLK_NOT_PRESENT	Indicates whether COMPLEX4CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present
[3]	COMPLEX3CLK_NOT_PRESENT	Indicates whether COMPLEX3CLK is present or not. • 0x0 – clock is present • 0x1 – clock is not present

Bits	Name	Description
[2]	COMPLEX2CLK_NOT_PRESENT	Indicates whether COMPLEX2CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[1]	COMPLEX1CLK_NOT_PRESENT	Indicates whether COMPLEX1CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present
[0]	COMPLEX0CLK_NOT_PRESENT	Indicates whether COMPLEX0CLK is present or not. • `0x0` – clock is present • `0x1` – clock is not present

7.3.6.29 PWR_CTRL_CONFIG, Core Manager and Power Control Configuration register

The power controller logical ID value is captured in this register. The value depends on the chosen configuration.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FC0`

Type

RO

Reset value

`0x0014_0000`

Bit descriptions

Table 7-102: PWR_CTRL_CONFIG bit descriptions

Bits	Name	Description
[31:16]	ID	Set to `0x0014`
[15:0]	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)

7.3.6.30 PID 4, Core Manager and Power Control Peripheral ID 4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FD0`

Type

RO

Reset value

`0x0000_0044`

Bit descriptions

Table 7-103: PID 4 bit descriptions

Bits	Name	Description
[31:8]	Reserved	Reserved, *Should-Be-Zero* (SBZ).
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as `0x4`. This means that the PIK occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many `0x7F` continuation characters occur in the manufacturer's identity code. These bits read back as `0x4`.

7.3.6.31 PID 5-7, Core Manager and Power Control Peripheral ID 5-7 registers

The PID5-7 registers are Reserved, *Read-As-Zero*, *Writes Ignored* (RAZ/WI).

7.3.6.32 PID 0, Core Manager and Power Control Peripheral ID 0 register

The PID0 register contains the first eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FE0`

Type

RO

Reset value

`0x0000_00B8`

Bit descriptions

Table 7-104: PID 0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	part_number_0	These bits read back as `0xB8`

7.3.6.33 PID 1, Core Manager and Power Control Peripheral ID 1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code and the second four bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FE4`

Type

RO

Reset value

`0x0000_00B0`

Bit descriptions

Table 7-105: PID 1 register bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. These bits read back as `0xB`.
[3:0]	part_number_1	Part Number. These bits read back as `0x0`

7.3.6.34 PID 2, Core Manager and Power Control Peripheral ID 2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FE8`

Type

RO

Reset value

`0x0000_0007`

Bit descriptions

Table 7-106: PID 2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	Revision	Identifies the revision of the base PIK. For revision r1p0, this field is set to `0x1`.
[3]	jedec_used	This indicates that the PIK uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as `0x1`
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.3.6.35 PID 3, Core Manager and Power Control Peripheral ID 3 register

The PID3 register contains the manufacturer revision number and a value that is incremented with each authorized customer modification.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FEC`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-107: PID 3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	RevAnd	This is set to `0x0`
[3:0]	mod_number	This is set to `0x0`

7.3.6.36 CID0, Core Manager and Power Control Component ID 0 register

The CID0 register contains segment 0 of the core manager and power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0FF0

Type

RO

Reset value

0x0000_000D

Bit descriptions

Table 7-108: CID 0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_0	These bits read back as 0x0D.

7.3.6.37 CID1, Core Manager and Power Control Component ID 1 register

The CID1 register contains segment 1 of the core manager and power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

0x0FF4

Type

RO

Reset value

0x0000_00F0

Bit descriptions

Table 7-109: CID 1 bit descriptions

Bits	Name	Description
[31;8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_1	These bits read back as 0xF0

7.3.6.38 CID2, Core Manager and Power Control Component ID 2 register

The CID2 register contains segment 2 of the core manager and power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FF8`

Type

RO

Reset value

`0x0000_0005`

Bit descriptions

Table 7-110: CID 2 bit descriptions

Bits	Name	Description
[31;8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_2	These bits read back as `0x05`

7.3.6.39 CID3, Core Manager and Power Control Component ID 3 register

The CID3 register contains segment 3 of the core manager and power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Core Manager and Power Control (CPU PIK) registers

Address offset

`0x0FFC`

Type

RO

Reset value

0x0000_00B1

Bit descriptions

Table 7-111: CID 3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_3	These bits read back as 0xB1

7.3.7 System Power Integration Kit (System PIK) registers

The System PIK registers enable configuration of clock settings for the system Power Integration Kit.

The System PIK is only mapped into the *System Control Processor* (SCP) memory map. For more information, see SCP memory map

The System PIK occupies a 64KB which is split into 4KB blocks as shown in the following table:

Table 7-112: System PIK address offsets

Offset	Name	Description
0x0000	PIK Control registers	Clock and pseudo static control signals for SYSTOP Clocks
0x1000	SYS-PPU0	Responsible for *Cache Coherent Mesh Network* (CMN) Logic P-Channel
0x2000-0x4FFF	-	Reserved
0x5000	SYS-PPU1	Responsible for SRAM Q-Channel
0x6000-0xFFFF	-	Reserved

The following table lists the System PIK register summary.

Table 7-113: System PIK register summary

Offset	Name	Type	Reset	Width	Description
0x000-0x7FC	-	-	-	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
0x800	PPUCLK_CTRL	RW	0x0000_0001	32-bit	System *Power Policy Unit* (PPU) Clock Control register
0x804	PPUCLK_DIV1	RW	0x0000_001F	32-bit	System PPU Clock Divider Control register
0x808-0x81C	-	-	-	-	Reserved, RAZ/WI
0x820	INTCLK_CTRL	RW	0x0000_0001	32-bit	Cache Coherent Interconnect Clock Control register
0x824	INTCLK_DIV1	RW	0x0000_001F	32-bit	Cache Coherent Interconnect Clock Divider Control register
0x828-0x82C	-	-	-	-	Reserved, RAZ/WI

Offset	Name	Type	Reset	Width	Description
0x830	TCU1CLK_CTRL	RW	0x0000_0001	32-bit	TCU1 Clock Control register
0x834	TCU1CLK_DIV1	RW	0x0000_001F	32-bit	TCU1 Clock Divider Control register
0x838	TCU2CLK_CTRL	RW	0x0000_0001	32-bit	TCU2 Clock Control register
0x83C	TCU2CLK_DIV1	RW	0x0000_001F	32-bit	TCU2 Clock Divider Control register
0x840	TCU3CLK_CTRL	RW	0x0000_0001	32-bit	TCU3 Clock Control register
0x844	TCU3CLK_DIV1	RW	0x0000_001F	32-bit	TCU3 Clock Divider Control register
0x848	TCU4CLK_CTRL	RW	0x0000_0001	32-bit	TCU4 Clock Control register
0x84C	TCU4CLK_DIV1	RW	0x0000_001F	32-bit	TCU4 Clock Divider Control register
0x850	GICCLK_CTRL	RW	0x0000_0001	32-bit	GIC Clock Control register
0x854	GICCLK_DIV1	RW	0x0000_001F	32-bit	GIC Clock Divider Control register
0x858-0x85C	-	-	-	-	Reserved, RAZ/WI
0x860	PCLKSCP_CTRL	RW	0x0000_0001	32-bit	*System Control Processor* (SCP) APB Clock Control register
0x864	PCLKSCP_DIV1	RW	0x0000_001F	32-bit	SCP APB Clock Divider Control register
0x868-0x86C	-	-	-	-	Reserved, RAZ/WI
0x870	SYSPERCLK_CTRL	RW	0x0000_0001	32-bit	System Peripheral Clock Control register
0x874	SYSPERCLK_DIV1	RW	0x0000_001F	32-bit	System Peripheral Clock Divider Control register
0x878-0x87C	-	-	-	-	Reserved, RAZ/WI
0x880	SCLK_CTRL	RW	0x0000_0001	32-bit	SCLK Clock Control register
0x884	SCLK_DIV1	RW	0x0000_001F	32-bit	SCLK Divider Control register
0x888-0x88C	-	-	-	-	Reserved, RAZ/WI
0x890	-	-	-	-	Reserved, RAZ/WI
0x894	-	-	-	-	Reserved, RAZ/WI
0x898-0x89C	-	-	-	-	Reserved, RAZ/WI
0x8A0	NS_UARTCLK_CTRL	RW	0x0000_0001	32-bit	Non-secure *Universal Asynchronous Receiver/Transmitter* (UART) Clock Control register
0x8A4	NS_UARTCLK_DIV1	RW	0x0000_001F	32-bit	Non-secure UART Clock Divider Control register
0x8A8	S_UARTCLK_CTRL	RW	0x0000_0001	32-bit	Secure UART Clock Control register
0x8AC	S_UARTCLK_DIV1	RW	0x0000_001F	32-bit	Secure UART Clock Divider Control register
0x8B0-0x9FC	-	-	-	-	Reserved, RAZ/WI
0x8B8-0x9FC	-	-	-	-	Reserved, RAZ/WI
0xA00	CLKFORCE_STATUS	RO	-	32-bit	System PIK Clock Force Status register
0xA04	CLKFORCE_SET	WO	-	32-bit	System PIK Clock Force Set register
0xA08	CLKFORCE_CLR	WO	0x0	32-bit	System PIK Clock Force Clear register
0xA0C-0xAFC	-	-	-	32-bit	Reserved, RAZ/WI
0xB08	-	-	-	32-bit	Reserved, RAZ/WI

Offset	Name	Type	Reset	Width	Description
0xB0C	SYSTOP_RST_DLY	RW	0x0000_0018	32-bit	Delay value for SYSTOPRESETn
0xB10-0xBFC	-	-	-	32-bit	Reserved, RAZ/WI
0xC00	-	-	-	32-bit	Reserved, RAZ/WI
0xC04-0xFBC	-	-	-	32-bit	Reserved, RAZ/WI
0xFC0	PWR_CTRL_CONFIG	RO	0x0021_0002	32-bit	System PIK Power Control Logic Configuration register
0xFD0	PID4	RO	0x44	32-bit	System PIK Peripheral ID 4 register
0xFD4	PID5	RO	0x00	32-bit	System PIK Peripheral ID 5 register
0xFD8	PID6	RO	0x00	32-bit	System PIK Peripheral ID 6 register
0xFDC	PID7	RO	0x00	32-bit	System PIK Peripheral ID 7 register
0xFE0	PID0	RO	0x00	32-bit	System PIK Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	32-bit	System PIK Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	32-bit	System PIK Peripheral ID 2 register
0xFEC	PID3	RO	0x00	32-bit	System PIK Peripheral ID 3 register
0xFF0	CID0	RO	0x0D	32-bit	System PIK Component ID 0 register
0xFF4	CID1	RO	0xF0	32-bit	System PIK Component ID 1 register
0xFF8	CID2	RO	0x05	32-bit	System PIK Component ID 2 register
0xFFC	CID3	RO	0xB1	32-bit	System PIK Component ID 3 register

7.3.7.1 PPUCLK_CTRL, System PPU Clock Control register

This register provides the ability to program the number of clock cycles between the PPUCLK not being required and the request to dynamically clock gate it. The clock source of the PPUCLK can also be programmed through this register.

This register shares the same bit descriptions as the following clock control registers:

- PCLKSCP_CTRL, SCP APB Clock Control register
- TCU<n>CLK_CTRL, TCU<n> Clock Control register
- SYSPERCLK_CTRL, System Peripheral Clock Control register
- GICCLK_CTRL, GIC Clock Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

0x800

Type

RW

Reset value

0x0000_0001

Bit descriptions

Table 7-114: PPUCLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. 0x0 – No cycles 0x1 – 1 cycle ... 0xFF – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source: • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.2 PPUCLK_DIV1, System PPU Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock to generate the required output clock. The current divider value can also be read out from this register.

This register shares the same bit descriptions as the following clock divider control registers:

- PCLKSCP_DIV1, SCP APB Clock Divider Control register
- TCU<n>CLK_DIV1, TCU<n> Clock Divider Control register
- SYSPERCLK_DIV1, System Peripheral Clock Divider Control register
- SYSPCLKDBG_DIV1, CPU Debug APB Completer Port Clock Divider Control register

- GICCLK_DIV1, GIC Clock Divider Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x804`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-115: PPUCLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, • CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	CLK_DIVn register requests a new clock divider value for the clock source selected by bit n, for example, • CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.3 INTCLK_CTRL, Cache Coherent Interconnect Clock Control register

This register provides the ability to program the number of clock cycles between the INTCLK not being required and the request to dynamically clock gate it. The clock source of the INTCLK can also be programmed through this register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x820`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-116: INTCLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source: 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – INTPLLCLK Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – INTPLLCLK Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.4 INTCLK_DIV1, Cache Coherent Interconnect Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock, INTPLLCLK, to generate the required output clock, INTCLK. The current divider value can also be read out from this register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x824`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-117: INTCLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n, for example: • CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	CLK_DIVn register requests a new clock divider value for the clock source selected by bit n, for example: • CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.5 TCU<n>CLK_CTRL, TCU<n> Clock Control register

These registers provide the ability to program the number of clock cycles between the TCU<n>CLK not being required and the request to dynamically clock gate it. The clock source of the TCU<n>CLK can also be programmed through this register.

This register shares the same bit descriptions as the following clock control registers:

- PPUCLK_CTRL, System PPU Clock Control register

- PCLKSCP_CTRL, SCP APB Clock Control register
- SYSPERCLK_CTRL, System Peripheral Clock Control register
- GICCLK_CTRL, GIC Clock Control register

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

See System Power Integration Kit (System PIK) registers

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-118: TCU<n>CLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source: • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.6 TCU<n>CLK_DIV1, TCU<n> Clock Divider Control register

These registers provide the ability to request a new clock divider value on the source clock to generate the rquired ouput clock. The current divider values can also be read out from these registers.

This register shares the same bit descriptions as the following clock divider control registers:

- PPUCLK_DIV1, System PPU Clock Divider Control register
- PCLKSCP_DIV1, SCP APB Clock Divider Control register
- SYSPERCLK_DIV1, System Peripheral Clock Divider Control register
- SYSPCLKDBG_DIV1, CPU Debug APB Completer Port Clock Divider Control register
- GICCLK_DIV1, GIC Clock Divider Control register

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

See System Power Integration Kit (System PIK) registers

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-119: TCU<n>CLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, • CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	CLK_DIVn register requests a new clock divider value for the clock source selected by bit n, for example, • CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.7 GICCLK_CTRL, GIC Clock Control register

This register provides the ability to program the number of clock cycles between the GICCLK not being required and the request to dynamically clock gate it. The clock source of the GICCLK can also be programmed through this register.

This register shares the same bit descriptions as the following clock control registers:

- PPUCLK_CTRL, System PPU Clock Control register
- PCLKSCP_CTRL, SCP APB Clock Control register
- TCU<n>CLK_CTRL, TCU<n> Clock Control register
- SYSPERCLK_CTRL, System Peripheral Clock Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x850`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-120: GICCLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles
[23:16]	-	Reserved

Bits	Name	Description
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source: • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.8 GICCLK_DIV1, GIC Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock (GICPLLCLK) to generate the required output clock (GICCLK). The current divider value can also be read out from this register.

This register shares the same bit descriptions as the following clock divider control registers:

- PPUCLK_DIV1, System PPU Clock Divider Control register
- PCLKSCP_DIV1, SCP APB Clock Divider Control register
- TCU<n>CLK_DIV1, TCU<n> Clock Divider Control register
- SYSPERCLK_DIV1, System Peripheral Clock Divider Control register
- SYSPCLKDBG_DIV1, CPU Debug APB Completer Port Clock Divider Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x854`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-121: GICCLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, • CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	CLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example, • CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.9 SYSPERCLK_CTRL, System Peripheral Clock Control register

This register provides the ability to program the number of clock cycles between the SYSPERCLK not being required and the request to dynamically clock gate it. The clock source of the SYSPERCLK can also be programmed through this register.

This register shares the same bit descriptions as the following clock control registers:

- PPUCLK_CTRL, System PPU Clock Control register
- PCLKSCP_CTRL, SCP APB Clock Control register
- TCU<n>CLK_CTRL, TCU<n> Clock Control register
- GICCLK_CTRL, GIC Clock Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x870`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-122: SYSPERCLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source: • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.10 SYSPERCLK_DIV1, System Peripheral Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock (SYSPLLCLK) to generate the required output clock (SYSPERCLK). The current divider value can also be read out from this register.

This register shares the same bit descriptions as the following clock divider control registers:

- PPUCLK_DIV1, System PPU Clock Divider Control register
- PCLKSCP_DIV1, SCP APB Clock Divider Control register
- TCU<n>CLK_DIV1, TCU<n> Clock Divider Control register
- SYSPCLKDBG_DIV1, CPU Debug APB Completer Port Clock Divider Control register
- GICCLK_DIV1, GIC Clock Divider Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x874`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-123: SYSPERCLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, • CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	CLK_DIVn register requests a new clock divider value for the clock source selected by bit n, for example, • CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.11 PCLKSCP_CTRL, SCP APB Clock Control register

This register provides the ability to program the number of clock cycles between the PCLKSCP not being required and the request to dynamically clock gate it. The clock source of the PCLKSCP can also be programmed through this register.

This register shares the same bit descriptions as the following clock control registers:

- PPUCLK_CTRL, System PPU Clock Control register
- TCU<n>CLK_CTRL, TCU<n> Clock Control register
- SYSPERCLK_CTRL, System Peripheral Clock Control register
- GICCLK_CTRL, GIC Clock Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x860`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-124: PCLKSCP_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source: • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.12 PCLKSCP_DIV1, SCP APB Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock to generate the required output clock. The current divider value can also be read out from this register.

This register shares the same bit descriptions as the following clock divider control registers:

- PPUCLK_DIV1, System PPU Clock Divider Control register
- TCU<n>CLK_DIV1, TCU<n> Clock Divider Control register
- SYSPERCLK_DIV1, System Peripheral Clock Divider Control register
- SYSPCLKDBG_DIV1, CPU Debug APB Completer Port Clock Divider Control register
- GICCLK_DIV1, GIC Clock Divider Control register

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x864`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-125: PCLKSCP_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, • CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	CLK_DIVn register requests a new clock divider value for the clock source selected by bit n, for example, • CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.13 SCLK_CTRL, DMC Clock Control register

This register provides the ability to program the number of clock cycles between the SCLK not being required and the request to dynamically clock gate it. The clock source of the SCLK can also be programmed through this register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x880`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-126: SCLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. • `0x0` – No cycles • `0x1` – 1 cycle ... • `0xFF` – 255 cycles
[23:17]	-	Reserved
16	SCLK_1XCLKBYPASSDIV2	Control bit for ClkMux on 1x and 2x clock after the divider, to bypass the divider in the SCLK clock selection. Default set to 1'b0. • 0 – div2 applied. SCLK1x is ½ the frequency of SCLK2x. (Default) • 1 – div2 bypassed. SCLK1x is the same frequency as SCLK2x.
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – DDRPLL Other values are Reserved

Bits	Name	Description
[7:0]	CLKSELECT	Selects the clock source • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – DDRPLL Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.14 SCLK_DIV1, SCLK Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock to generate the required output clock. The current divider value can also be read out from this register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x884`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-127: SCLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example: • GICCLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	The GICCLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example: • GICCLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.15 NS_UARTCLK_CTRL, Non-secure UART Clock Control register

This register provides the ability to program the number of clock cycles between the NS_UARTCLK not being required and the request to dynamically clock gate it. The clock source of the NS_UARTCLK can also be programmed through this register.

This register shares the same bit descriptions as the S_UARTCLK_CTRL register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x8A0`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-128: NS_UARTCLK_CTRL descriptions

Bits	Name	Description
[31:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.16 S_UARTCLK_CTRL, Secure UART Clock Control register

This register provides the ability to program the number of clock cycles between the S_UARTCLK not being required and the request to dynamically clock gate it. The clock source of the S_UARTCLK can also be programmed through this register.

This register shares the same bit descriptions as the NS_UARTCLK_CTRL register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x8A8`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-129: S_UARTCLK_CTRL bit descriptions

Bits	Name	Description
[31:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.7.17 NS_UARTCLK_DIV1, Non-secure UART Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock to generate the required output clock. The current divider value can also be read out from this register.

This register shares the same bit descriptions as the S_UARTCLK_DIV1 register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x8A4`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-130: NS_UARTCLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n, for example: • GICCLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	The GICCLK_DIVn register requests a new clock divider value for the clock source selected by bit n, for example: • GICCLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.18 S_UARTCLK_DIV1, Secure UART Clock Divider Control register

This register provides the ability to request a new clock divider value on the source clock to generate the required output clock. The current divider value can also be read out from this register.

This register shares the same bit descriptions as the NS_UARTCLK_DIV1 register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0x8AC`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-131: S_UARTCLK_DIV1 bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n, for example: • GICCLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	The GICCLK_DIVn register requests a new clock divider value for the clock source selected by bit n, for example: • GICCLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.3.7.19 CLKFORCE_STATUS, System PIK Clock Force Status register

The CLKFORCE_STATUS register records whether the dynamic clock gating associated with that clock is enabled (0) or disabled (1).

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xA00`

Type

RO

Bit descriptions

If a bit reads back as one, then the dynamic clock gating associated with the clock is disabled, otherwise it is enabled.

Table 7-132: CLKFORCE_STATUS bit descriptions

Bits	Name	Description
[31:13]	-	Reserved
[13]	GICCLKFORCE	Clock force status for GICCLK
[12:9]	TCUCLKFORCE[3:0]	Clock force status for TCUCLK1-TCUCLK4
[8]	SYSPCLKDBGFORCE	Clock force status for SYSPCLKDBG
[7]	DMCCLKFORCE	Clock force status for DMCCLK
[6]	SYSPERCLKFORCE	Clock force status for SYSPERCLKFORCE
[5]	PCLKSCPFORCE	Clock force status for PCLKSCP
[4:3]	-	Reserved
[2]	INTCLKFORCE	Clock force status for INTCLK
[1]	-	Reserved
[0]	PPUCLKFORCE	Clock force status for PPUCLK

7.3.7.20 CLKFORCE_SET, System PIK Clock Force Set register

Writing a 1 to a bit within the CLKFORCE_SET register disables any dynamic hardware clock gating, while writing 0 to a bit is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

0xA04

Type

WO

Bit descriptions

The bit allocation is the same as the CLKFORCE_STATUS register.

Table 7-133: CLKFORCE_SET bit descriptions

Bits	Name	Description
[31:13]	-	Reserved
[13]	GICCLKFORCE	Clock force set for GICCLK
[12:9]	TCUCLKFORCE[3:0]	Clock force set for TCUCLK1-TCUCLK4
[8]	SYSPCLKDBGFORCE	Clock force set for SYSPCLKDBG
[7]	DMCCLKFORCE	Clock force set for DMCCLK
[6]	SYSPERCLKFORCE	Clock force set for SYSPERCLKFORCE
[5]	PCLKSCPFORCE	Clock force set for PCLKSCP
[4:3]	-	Reserved
[2]	INTCLKFORCE	Clock force set for INTCLK
[1]	-	Reserved
[0]	PPUCLKFORCE	Clock force set for PPUCLK

7.3.7.21 CLKFORCE_CLR, System PIK Clock Force Clear register

Writing a 1 to a bit within the CLKFORCE_CLR register disables any dynamic hardware clock gating, while writing 0 to a bit is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xA08`

Type

WO

Reset value

`0x0`

Bit descriptions

The bit allocation is the same as the CLKFORCE_STATUS register.

Table 7-134: CLKFORCE_CLR bit descriptions

Bits	Name	Description
[31:13]	-	Reserved
[13]	GICCLKFORCE	Clock force clear for GICCLK
[12:9]	TCUCLKFORCE[3:0]	Clock force clear for TCUCLK1-TCUCLK4
[8]	SYSPCLKDBGFORCE	Clock force clear for SYSPCLKDBG
[7]	DMCCLKFORCE	Clock force clear for DMCCLK
[6]	SYSPERCLKFORCE	Clock force clear for SYSPERCLKFORCE
[5]	PCLKSCPFORCE	Clock force clear for PCLKSCP
[4:3]	-	Reserved
[2]	INTCLKFORCE	Clock force clear for INTCLK
[1]	-	Reserved
[0]	PPUCLKFORCE	Clock force clear for PPUCLK

7.3.7.22 SYSTOP_RST_DLY, SYSTOP Reset Delay register

This register controls the delayed release of the nSYSTOPRESET reset. We advise you do not modify this value.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

0xB0C

Type

RW

Reset value

0x0000_0018

Bit descriptions

Table 7-135: : SYSTOP_RST_DLY bit descriptions

Bits	Name	Description
[31:6]	-	Reserved
[5:0]	RST_DLY	Delay value for nSYSTOPRESET. The reset value is 6'h18.

7.3.7.23 PWR_CTRL_CONFIG, System PIK Power Control Logic Configuration register

The power controller logical ID value is captured in this register. The value depends on the chosen configuration.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

0xFC0

Type

RO

Reset value

0x0021_0002

Bit descriptions

Table 7-136: PWR_CTRL_CONFIG bit descriptions

Bits	Name	Description
[31:16]	-	ID. This field is set to 0x0024.

Bits	Name	Description
[15:4]	-	Reserved
[3:0]	no_of_ppu	Defines the number of *Power Policy Units* (PPUs) in the power control logic. This value is set to indicate number of PPUs. The value is dependent on the number PPUs implemented in the subsystem. This reads back as 2.

7.3.7.24 PID4, System PIK Peripheral ID 4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xFD0`

Type

RO

Reset value

`0x44`

Bit descriptions

Table 7-137: PID4 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as `0x4`. This means that the PIK occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many `0x7F` continuation characters occur in the manufacturer's identity code. These bits read back as `0x4`.

7.3.7.25 PID5-7, System PIK Peripheral ID 5-7 register

The System PIK Peripheral ID5-7 registers are Reserved, *Read-As-Zero*, *Writes Ignored* (RAZ/WI).

7.3.7.26 PID0, System PIK Peripheral ID 0 register

The PID0 register contains the first eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xFE0`

Type

RO

Reset value

`0x00`

Bit descriptions

Table 7-138: PID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	part_number_0	These bits read back as `0xB8`.

7.3.7.27 PID1, System PIK Peripheral ID 1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code and the second four bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

0xFE4

Type

RO

Reset value

0xB0

Bit descriptions

Table 7-139: PID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. These bits read back as 0xB.
[3:0]	part_number_1	Part Number. These bits read back as 0x0.

7.3.7.28 PID2, System PIK Peripheral ID 2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

0xFE8

Type

RO

Reset value

0x0B

Bit descriptions

Table 7-140: PID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).

Bits	Name	Description
[7:4]	Revision	Identifies the revision of the base PIK. For revision r0p0, this field is set to `0x0`.
[3]	jedec_used	This indicates that the PIK uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as `0x1`.
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.3.7.29 PID3, System PIK Peripheral ID 3 register

The PID3 register contains the manufacturer silicon revision number and a value that is incremented with each authorized customer modification.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xFEC`

Type

RO

Reset value

`0x00`

Bit descriptions

Table 7-141: PID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	RevAnd	This is set to `0x0`
[3:0]	mod_number	This is set to `0x0`

7.3.7.30 CID0, System PIK Component ID 0 register

The CID0 register contains segment 0 of the system PIK component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xFF0`

Type

RO

Reset value

`0x0D`

Bit descriptions

Table 7-142: CID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_0	These bits read back as `0x0D`

7.3.7.31 CID1, System PIK Component ID 1 register

The CID1 register contains segment 1 of the system PIK component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-143: CID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_1	These bits read back as `0xF0`

7.3.7.32 CID2, System PIK Component ID 2 register

The CID2 register contains segment 2 of the system PIK component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-144: CID2 register bit assignment

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_2	These bits read back as `0x05`

7.3.7.33 CID3, System PIK Component ID 3 register

The CID3 register contains segment 3 of the system PIK component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System Power Integration Kit (System PIK) registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-145: CID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_3	These bits read back as `0xB1`

7.3.8 Debug Power Control Logic (Debug PIK) registers

The Debug Power Control Logic registers provide information about the power settings for debug in a subsystem.

Arm Total Compute 2022 Reference Design (RD-TC22) supports *Arm® Power Policy Unit Architecture Specification, version 1.1* for Debug PPUs.

The Debug PIK occupies a 64KB block which is split into 4KB blocks as shown in the following table.

Table 7-146: Debug PIK address offset

Offset	Description	Notes
`0x0000`	Debug PIK Control Registers	Clocks and debug control registers
`0x1000`	DBGPIK-PPU	Responsible for Debug Top logic
`0x2000-0x4FFF`	-	Reserved

For more information, see the *Arm® Power Policy Unit Architecture Specification, version 1.1*.

The following table summarizes the Debug PIK registers.

Table 7-147: Debug Power Control Logic register summary

Offset	Name	Type	Reset	Width	Description
`0x000`	DBG_PWR_REQ_ST0	RO	-	32-bit	Debug Power Request Status 0 register

Offset	Name	Type	Reset	Width	Description
0x004	DBG_PWR_REQ_ST1	RO	-	32-bit	Debug Power Request Status 1 register
0x008	DBG_PWR_REQ_ST2	RO	-	32-bit	Debug Power Request Status 2 register
0x00C	DBG_PWR_REQ_ST3	RO	-	32-bit	Debug Power Request Status 3 register
0x010	DBG_PWR_ACK0	RW	0x0	32-bit	Debug Power Acknowledge 0 register
0x014	DBG_PWR_ACK1	RW	0x0	32-bit	Debug Power Acknowledge 1 register
0x018	DBG_PWR_ACK2	RW	0x0	32-bit	Debug Power Acknowledge 2 register
0x01C	DBG_PWR_ACK3	RW	0x0	32-bit	Debug Power Acknowledge 3 register
0x020	DBG_RST_REQ_ST	RO	-	32-bit	Debug Power Request Status register
0x024	DBG_RST_ACK	RW	0x0	32-bit	Debug Reset Acknowledge register
0x028 – 0x02C	Reserved	RAZ/WI	-	-	-
0x030	SYS_PWR_REQ_ST0	RO	-	32-bit	System Power Request Status 0 register
0x034	SYS_PWR_REQ_ST1	RO	-	32-bit	System Power Request Status 1 register
0x038	SYS_PWR_REQ_ST2	RO	-	32-bit	System Power Request Status 2 register
0x03C	SYS_PWR_REQ_ST3	RO	-	32-bit	System Power Request Status 3 register
0x040	SYS_PWR_ACK0	RW	0x0	32-bit	System Power Acknowledge 0 register
0x044	SYS_PWR_ACK1	RW	0x0	32-bit	System Power Acknowledge 1 register
0x048	SYS_PWR_ACK2	RW	0x0	32-bit	System Power Acknowledge 2 register
0x04C	SYS_PWR_ACK3	RW	0x0	32-bit	System Power Acknowledge 3 register
0x050	SYS_RST_REQ_ST	RO	-	32-bit	System Power Request Status register
0x054	SYS_RST_ACK	RW	0x0	32-bit	System Power Acknowledge register
0x058	DBGACK	RO	0x0	32-bit	Debug ACK per *Processing Element* (PE) register
0x05C	Reserved	RAZ/WI	-	-	-
0x060	DEBUG_CONFIG	RW	0x0	32-bit	Debug Configuration register
0x070 – 0x7FC	Reserved	RAZ/WI	-	-	-
0x800 – 0x80C	Reserved	RAZ/WI	-	-	-
0x810	TRACECLK_CTRL	RW	0x1	32-bit	Trace Clock Control register
0x814	TRACECLK_DIV1	RW	0x1F	32-bit	Trace Clock Divider register
0x818 – 0x81C	Reserved	RAZ/WI	-	-	-
0x820	PCLKDBG_CTRL	RW	0x1	32-bit	Debug APB Clock Control register
0x824	PCLKDBG_DIV1	RW	0x1F	32-bit	Debug APB Clock Divider register
0x828 – 0x82C	Reserved	RAZ/WI	-	-	-
0x830	DBGCLK_CTRL	RW	0x1	32-bit	Debug Clock Control register
0x834	DBGCLK_DIV1	RW	0x1F	32-bit	Debug Clock Divider register
0x838 – 0x83C	Reserved	RAZ/WI	-	-	-

Offset	Name	Type	Reset	Width	Description
0x840-0x9FC	Reserved	RAZ/WI	-	-	-
0xA00	CLKFORCE_STATUS	RO	-	32-bit	Clock Force Status register
0xA04	CLKFORCE_SET	WO	-	32-bit	Clock Force Set register
0xA08	CLKFORCE_CLR	WO	-	32-bit	Clock Force Clear register
0xB00	DBG_PWR_REQ_INT_ST0	RW1C	0x0	32-bit	Debug Power Request Init Status 0 register
0xB04	DBG_PWR_REQ_INT_ST1	RW1C	0x0	32-bit	Debug Power Request Init Status 1 register
0xB08	DBG_PWR_REQ_INT_ST2	RW1C	0x0	32-bit	Debug Power Request Init Status 2 register
0xB0C	DBG_PWR_REQ_INT_ST3	RW1C	0x0	32-bit	Debug Power Request Init Status 3 register
0xB10	DBG_RST_REQ_INT_ST	RW1C	0x0	32-bit	Debug Reset Request Init Status 2 register
0xB14 – 0xB1C	Reserved	RAZ/WI	-	-	-
0xB20	SYS_PWR_REQ_INT_ST0	RW1C	0x0	32-bit	System Power Request Init Status 0 register
0xB24	SYS_PWR_REQ_INT_ST1	RW1C	0x0	32-bit	System Power Request Init Status 1 register
0xB28	SYS_PWR_REQ_INT_ST2	RW1C	0x0	32-bit	System Power Request Init Status 2 register
0xB2C	SYS_PWR_REQ_INT_ST3	RW1C	0x0	32-bit	System Power Request Init Status 3 register
0xB30	SYS_RST_REQ_INT_ST	RW1C	0x0	32-bit	System Power Request Init Status register
0xB40	DEBUG_RCOV	RW	0x0	32-bit	Debug RCOV Control register
0xFBC	CAP0	RO	IMPLEMENTATION DEFINED	32-bit	Debug Capability register
0xFC0	PWR_CTRL_CONFIG	RO	0x00330001	32-bit	Debug Power Control Logic Configuration register
0xFD0	PID4	RO	0x44	32-bit	Debug Peripheral ID 4 register
0xFD4	PID5	RO	0x00	32-bit	Debug Peripheral ID 5 register
0xFD8	PID6	RO	0x00	32-bit	Debug Peripheral ID 6 register
0xFDC	PID7	RO	0x00	32-bit	Debug Peripheral ID 7 register
0xFE0	PID0	RO	0xB8	32-bit	Debug Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	32-bit	Debug Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	32-bit	Debug Peripheral ID 2 register
0xFEC	PID3	RO	0x00	32-bit	Debug Peripheral ID 3 register
0xFF0	CID0	RO	0x0D	32-bit	Debug Component ID 0 register
0xFF4	CID1	RO	0xF0	32-bit	Debug Component ID 1 register
0xFF8	CID2	RO	0x05	32-bit	Debug Component ID 2 register
0xFFC	CID3	RO	0xB1	32-bit	Debug Component ID 3 register
0x1000 – 0x1FFC	DEBUG_PPU	-	-	-	Debug PPU Configuration-dependent registers

7.3.8.1 DBG_PWR_REQ_ST0-3, Debug Power Request Status 0-3 registers

These registers are used to handle powerup and reset requests from an external debugger.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x000

Type

RO

Reset value

-

Bit descriptions

Table 7-148: DBG_PWR_REQ_ST0-3 bit descriptions

Bits	Name	Description
[31:N]	-	Reserved
[N-1:0]	DBGRESETREQ_ST	The status of the CDBGRESETREQx signal from the applications processor *Debug Access Port* (DAP).

The number of bits implemented, N, is equal to CAP.NUM_CDBGRST. If N=0 then the entire register is **RAZ/WI**.

7.3.8.2 DBG_PWR_ACK0-3, Debug Power Acknowledge 0-3 registers

These registers are used to handle powerup and reset requests from an external debugger.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x010 – 0x01C

Type

RW

Reset value

0x0

Bit descriptions

Table 7-149: DBG_PWR_ACK0-3 bit descriptions

Bit	Name	Description
[127:N]	-	Reserved
[N-1:0]	DBGPWRACK	Set the value of the CDBGPWRUPACKx.

Where, N is the number of bits implemented, which is equal to CAP.NUM_CDBGPWR.

- Bits [31:0] are in DBG_PWR_ACK0
- Bits [63:32] are in DBG_PWR_ACK1
- Bits [64:95] are in DBG_PWR_ACK2
- Bits [96:127] are in DBG_PWR_ACK3

7.3.8.3 DBG_RST_REQ_ST, Debug Reset Request Status register

This register is used to handle powerup and reset requests from an external debugger.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x000

Type

RO

Reset value

-

Bit descriptions

Table 7-150: DBG_RST_REQ_ST bit descriptions

Bit	Name	Description
[31:N]	-	Reserved
[N-1:0]	DBGRESETREQ_ST	Status of the CDBGRESETREQx.

Where, N is the number of bits implemented, which is equal to CAP.NUM_CDBGRST. If N=0 then the entire register is **RAZ/WI**.

7.3.8.4 DBG_RST_ACK, Debug Reset Acknowledge register

This register is used to handle powerup and reset requests from an external debugger.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x024

Type

RW

Reset value

0x0

Bit descriptions

Table 7-151: DBG_RST_ACK bit descriptions

Bit	Name	Description
[31:N]	-	Reserved
[N-1:0]	DBGRESETACK	Set the value of the CDBGRESETACKx.

Where, N is the number of bits implemented, which is equal to CAP.NUM_CDBGRST. If N=0 then the entire register is **RAZ/WI**.

7.3.8.5 SYS_PWR_REQ_ST0-3, System Power Request Status 0-3 registers

These registers are used to handle powerup and reset requests from an external debugger.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0x030 – 0x03C`

Type

RO

Reset value

-

Bit descriptions

Table 7-152: SYS_PWR_REQ_ST0-3 bit descriptions

Bit	Name	Description
[127:N]	-	Reserved
[N-1:0]	SYSPWRREQ_ST	Status of the CSYSPWRUPREQx.

Where, N is the number of bits implemented, which is equal to CAP.NUM_CSYSPWR.

- Bits [31:0] are in DBG_SYS_REQ_ST0
- Bits[63:32] are in DBG_SYS_REQ_ST1
- Bits [64:95] are in DBG_SYS_REQ_ST2
- Bits [96:127] are in DBG_SYS_REQ_ST3

7.3.8.6 SYS_PWR_ACK0-3, System Power Acknowledge 0-3 registers

These registers are used to handle powerup and reset requests from an external debugger.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x040 – 0x04C

Type

RW

Reset value

0x0

Bit descriptions

Table 7-153: SYS_PWR_ACK0-3 bit descriptions

Bit	Name	Description
[127:N]	-	Reserved
[N-1:0]	SYSPWRACK	Set the value of the CSYSPWRUPACKx.

Where, N is the number of bits implemented, which is equal to CAP.NUM_CSYSPWR.

- Bits [31:0] are in SYS_PWR_ACK0
- Bits[63:32] are in SYS_PWR_ACK1
- Bits [64:95] are in SYS_PWR_ACK2
- Bits [96:127] are in SYS_PWR_ACK3

7.3.8.7 SYS_RST_REQ_ST, System Reset Request Status register

This register is used to handle powerup and reset requests from an external debugger.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x050

Type

RO

Reset value

-

Bit descriptions

Table 7-154: SYS_RST_REQ_ST bit descriptions

Bit	Name	Description
[31:N]	-	Reserved
[N-1:0]	SYSRSTREQ_ST	Status of the CSYSRSTREQx.

Where, N is the number of bits implemented, which is equal to CAP.NUM_CSYSRST.

7.3.8.8 SYS_RST_ACK, System Reset Acknowledge register

This register is used to handle powerup and reset requests from an external debugger.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x054

Type

RW

Reset value

0x0

Bit descriptions

Table 7-155: SYS_RST_ACK bit descriptions

Bit	Name	Description
[31:N]	-	Reserved
[N-1:0]	SYSRSTACK	Set the value of the CSYSRSTACKx.

Where, N is the number of bits implemented, which is equal to CAP.NUM_CSYSRST. If N=0 then the entire register is **RAZ/WI**.

7.3.8.9 DBGACK, Debug ACK per PE register

This register is used to handle powerup and reset requests from an external debugger.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x058

Type

RO

Reset value

0x0

Bit descriptions

Table 7-156: DBGACK bit descriptions

Bits	Name	Description
[31:8]	-	Reserved
[7]	PE0_DBGACK	Status of DBCACK from PE7
[6]	PE0_DBGACK	Status of DBCACK from PE6
[5]	PE0_DBGACK	Status of DBCACK from PE5
[4]	PE0_DBGACK	Status of DBCACK from PE4
[3]	PE0_DBGACK	Status of DBCACK from PE3
[2]	PE0_DBGACK	Status of DBCACK from PE2
[1]	PE0_DBGACK	Status of DBCACK from PE1
[0]	PE0_DBGACK	Status of DBCACK from PE0

7.3.8.10 DEBUG_CONFIG, Debug Configuration register

This register controls the runtime debug configuration.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0x008

Type

RW

Reset value

0x0

Bit descriptions

Table 7-157: DEBUG_CONFIG bit descriptions

Bits	Name	Description
[31:1]	-	Reserved
[0]	DBGCONNECTED	Drives the DBGCONNECTED signal inputs into all processor clusters.

7.3.8.11 Debug Clock Control registers (TRACECLK_CTRL, DBGCLK_CTRL, PCLKDBG_CTRL)

These registers provide the ability to program the number of clock cycles between the output clock not being required and the request to dynamically clock gate it. The clock source of the clock can also be programmed through this register.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

RW

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

Table 7-158: Debug Clock Control registers bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.8.12 Debug Clock Divider Control registers (TRACECLK_DIV1, DBGCLK_DIV1, PCLKDBG_DIV1)

These registers provide the ability to request a new clock divider value on the source clock to generate the required output clock. The current divider value can also be read out from this register.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

RW

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

Table 7-159: Debug Clock Divider Control registers bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV_CUR + 1. For example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	CLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example, CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV + 1. For example, setting a value of 0 indicates a divider value of 1.

7.3.8.13 CLKFORCE_STATUS, Debug Clock Force Status register

The CLKFORCE_STATUS register records whether the dynamic clock gating associated with that clock is enabled (0) or disabled (1).

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

RO

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

If a bit reads back as 1, then the dynamic clock gating associated with the clock is disabled, otherwise it is enabled.

Table 7-160: CLKFORCE_STATUS bit descriptions

Bits	Name	Description
[31:4]	-	Reserved
[3]	DBGCLKFORCE	Clock force for DBGCLK.
[2]	PCLKDBGFORCE	Clock force for PCLKDBG.
[1:0]	-	Reserved

7.3.8.14 CLKFORCE_SET, Debug Clock Force SET register

Writing a 1 to a bit within the CLKFORCE_SET register disables any dynamic hardware clock gating for that respective clock, while writing 0 to a bit is ignored.

The CLFORCE_SET register shares the same bit assignment as the CLKFORCE_STATUS register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

WO

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

Table 7-161: CLKFORCE_SET bit descriptions

Bits	Name	Description
[31:4]	-	Reserved
[3]	DBGCLKFORCE	Clock force for DBGCLK.
[2]	PCLKDBGFORCE	Clock force for PCLKDBG.
[1:0]	-	Reserved

7.3.8.15 CLKFORCE_CLR, Debug Clock Force Clear register

Writing a 1 to a bit within the CLKFORCE_CLR register enables the dynamic hardware clock gating for that respective clock, while writing 0 to a bit is ignored.

The CLFORCE_CLR register shares the same bit assignment as the CLKFORCE_STATUS register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

WO

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

Table 7-162: CLKFORCE_CLR bit descriptions

Bits	Name	Description
[31:4]	-	Reserved
[3]	DBGCLKFORCE	Clock force for DBGCLK.
[2]	PCLKDBGFORCE	Clock force for PCLKDBG.
[1:0]	-	Reserved

7.3.8.16 Debug and System Power Request Init Status registers (DBG_PWR_REQ_INT_ST0-3, SYS_PWR_REQ_INT_ST0-3)

These registers indicate the status of the CDBGPWRUPREQ and SYSPWRREQ interrupts. The interrupt is cleared by writing 1 to the respective register.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

RW1C

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

Table 7-163: Debug and System Power Request Init Status registers bit descriptions

Bits	Name	Description
[31:N]	-	Reserved
[N-1:0]	CDBGPWRUPREQ_INT	This bit is set on any edge of the CDBGPWRUPREQx signal. Writing 1 to this bit clears it.

The number of bits implemented, N, is equal to CAP.NUM_CDBGPWR.

- Bits [31:0] are in DBG_PWR_REQ_INT_ST0 and SYS_PWR_REQ_INT_ST0
- Bits [63:32] are in DBG_PWR_REQ_INT_ST1 and SYS_PWR_REQ_INT_ST1
- Bits [64:95] are in DBG_PWR_REQ_INT_ST2 and SYS_PWR_REQ_INT_ST2
- Bits [96:127] are in DBG_PWR_REQ_INT_ST3 and SYS_PWR_REQ_INT_ST3

7.3.8.17 Debug and System Reset Request Init Status registers (DBG_RST_REQ_INT_ST, SYS_RST_REQ_INT_ST)

These registers indicate the status of the CDBGRESETREQ and SYSRSTREQ interrupts. The interrupts are cleared by writing 1 to the respective register.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

RW1C

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

Table 7-164: Debug and System Reset Request Init Status registers bit descriptions

Bits	Name	Description
[31:N]	-	Reserved
[N-1:0]	CDBGRESETREQ_INT	This bit is set on any edge of the CDBGRESETREQx signal. Writing 1 to this bit clears it.

The number of bits implemented, N, is equal to CAP.NUM_CDBGRST. If N=0 then the entire register is **RAZ/WI**.

7.3.8.18 DEBUG_RCOV, Debug RCOV Control register

This section describes the Debug RCOV Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xB40`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-165: DEBUG_RCOV bit descriptions

Bits	Name	Description
[31:1]	-	Reserved
[0]	DEBUG_RCOV	Set to transition one or more core *Power Policy Units* (PPUs) from the DBG_RCOV power mode to the ON state

7.3.8.19 CAP, Debug Capabilities register

This section describes the Debug Capabilities register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xFBC`

Type

RO

Reset value

IMPLEMENTATION DEFINED

Bit descriptions

Table 7-166: CAP bit descriptions

Bit	Name	Description
[31:28]	-	Reserved
[27:22]	NUM_CSYSRST	Number of system reset requests
[21:16]	NUM_CDBGRST	Number of debug domain reset requests
[15:8]	NUM_CSYSPWR	Number of system power domain power up requests
[7:0]	NUM_CDBGPWR	Number of debug power domain power up requests

7.3.8.20 PWR_CTRL_CONFIG, Debug Power Control Logic Configuration register

This section describes the Debug Chain Power Control Logic Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0xFC0

Type

RO

Reset value

0x00330001

Bit descriptions

Table 7-167: PWR_CTRL_CONFIG bit descriptions

Bits	Name	Description
[31:16]	-	POWER CONTROL LOGIC_ID. This field is set to 0x0033.
[15:4]	-	Reserved
[3:0]	No_of_ppu	Defines the number of *Power Policy Units* (PPUs) in the POWER CONTROL LOGIC. This value is set to 0x1 to indicate one PPU.

7.3.8.21 PID4, Debug Power Control Peripheral ID 4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0xFD0

Type

RO

Reset value

0x44

Bit descriptions

Table 7-168: PID4 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).

Bits	Name	Description
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as `0x4`. This means that the POWER CONTROL LOGIC occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many `0x7F` continuation characters occur in the manufacturer's identity code. These bits read back as `0x4`.

7.3.8.22 PID5-7, Debug Peripheral ID 5-7 register

The Peripheral ID 5-7 registers are Reserved, *Read-As-Zero*, *Writes Ignored* (RAZ/WI).

7.3.8.23 PID0, Debug Power Control Peripheral ID 0 register

The PID0 register contains the first eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xFE0`

Type

RO

Reset value

`0xB8`

Bit descriptions

Table 7-169: PID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	part_number_0	These bits read back as `0xB8`

7.3.8.24 PID1, Debug Power Control Peripheral ID 1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code and the second four bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xFE4`

Type

RO

Reset value

`0xB0`

Bit descriptions

Table 7-170: PID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. See the JEP106, Standard Manufacturer's Identification Code.
[3:0]	part_number_1	These bits read back as `0x0`

7.3.8.25 PID2, Debug Power Control Peripheral ID 2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0xFE8

Type

RO

Reset value

0x0B

Bit descriptions

Table 7-171: PID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	Revision	Identifies the revision of the base POWER CONTROL LOGIC. For revision r0p0, this field is set to 0x0.
[3]	jedec_used	This indicates that the POWER CONTROL LOGIC uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.3.8.26 PID3, Debug Power Control Peripheral ID 3 register

The PID3 register contains the manufacturer silicon revision number and a value that is incremented with each authorized customer modification.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

0xFEC

Type

RO

Reset value

0x00

Bit descriptions

Table 7-172: PID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)

Bits	Name	Description
[7:4]	RevAnd	The top level RTL provides a 4-bit input, ECOREVNUM, that is normally tied LOW and provides a read value of `0x0`. When silicon is available, and if metal fixes are necessary, the manufacturer can modify the tie-offs to indicate a revision of the silicon.
[3:0]	mod_number	This is set to `0x0`

7.3.8.27 CID0, Debug Power Control Component ID 0 register

The CID0 register contains segment 0 of the debug chain power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xFF0`

Type

RO

Reset value

`0x0D`

Bit descriptions

Table 7-173: CID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_0	These bits read back as `0x0D`

7.3.8.28 CID1, Debug Power Control Component ID 1 register

The CID1 register contains segment 1 of the debug chain power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-174: CID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_1	These bits read back as `0xF0`

7.3.8.29 CID2, Debug Power Control Component ID 2 register

The CID2 register contains segment 2 of the debug chain power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-175: CID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_2	These bits read back as `0x05`

7.3.8.30 CID3, Debug Power Control Component ID 3 register

The CID3 register contains segment 3 of the debug chain power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-176: CID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_3	These bits read back as `0xB1`

7.3.8.31 DEBUG_PPU, Debug PPU Configuration-dependent registers (PPU_IDR0 and PPU_IDR1)

This section describes the Debug Power Policy Unit registers, PPU_IDR0 and PPU_IDR1.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug Power Control Logic (Debug PIK) registers

Address offset

See Debug Power Control Logic (Debug PIK) registers

Type

RW

Reset value

See Debug Power Control Logic (Debug PIK) registers

Bit descriptions

Table 7-177: Debug PPU_IDR0 bit description

Bits	Name	Description
[31:30]	-	Reserved, SBZ
[29]	DYN_WRM_RST_SPT	0x0
[28]	DYN_ON_SPT	0x0
[27]	DYN_FNC_RET_SPT	0x0
[26]	DYN_FULL_RET_SPT	0x0
[25]	DYN_MEM_OFF_SPT	0x0
[24]	DYN_LGC_RET_SPT	0x0
[23]	DYN_MEM_RET_EMU_SPT	0x0
[22]	DYN_MEM_RET_SPT	0x0
[21]	DYN_OFF_EMU_SPT	0x0
[20]	DYN_OFF_SPT	0x0
[19]	-	Reserved, SBZ
[18]	STA_DBG_RECOV_SPT	0x0
[17]	STA_WRM_RST_SPT	0x1
[16]	STA_ON_SPT	0x1
[15]	STA_FNC_RET_SPT	0x0
[14]	STA_FULL_RET_SPT	0x0
[13]	STA_MEM_OFF_SPT	0x0
[12]	STA_LGC_RET_SPT	0x0
[11]	STA_MEM_RET_EMU_SPT	0x0
[10]	STA_MEM_RET_SPT	0x0
[9]	STA_OFF_EMU_SPT	0x0
[8]	STA_OFF_SPT	0x1
[7:4]	NUM_OPMODE	0x0
[3:0]	DEVCHAN	0x1

Table 7-178: Debug PPU_IDR1 bit description

Bits	Name	Description
[31:11]	-	Reserved, SBZ
[12]	OFF_MEM_RET_TRANS	0x0
[11]	-	Reserved, SBZ
[10]	OP_ACTIVE	0x0
[9]	STA_POLICY_OP_IRQ_SPT	0x0
[8]	STA_POLICY_PWR_IRQ_SPT	0x0
[7]	-	Reserved, SBZ
[6]	FUNC_RET_RAM_REG	0x0
[5]	FULL_RET_RAM_REG	0x0
[4]	MEM_RET_RAM_REG	0x0
[3]	-	Reserved, SBZ
[2]	LOCK_SPT	0x0
[1]	SW_DEV_DEL_SPT	0x1
[0]	PWR_MODE_ENTRY_DEL_SPT	0x1

7.3.9 GPU Power Control Logic (GPU PIK) registers

The GPU Power Control Logic registers occupy 64KB which is split into 4KB blocks as shown in the following table.

Table 7-179: GPU PIK address offset

Offset	Description	Notes
0x0000	GPU PIK Control Registers	Clocks and GPU control registers
0x1000	GPU-PPU0	Responsible for GPUTOP logic
0x2000-0xFFFF	-	Reserved

The following table lists the GPU power control logic register summary.

Table 7-180: GPU power control logic register summary

Offset	Name	Type	Reset	Width	Description
0x000 – 0x7FC	-	RO	-	32-bit	Reserved, *Read-As-Zero*, *Writes Ignored* (RAZ/WI)
0x800 – 0x80C	-	RO	-	-	Reserved, RAZ/WI
0x810	GPUCLK_CTRL	RW	0x0000_0001	32-bit	GPU Clock Control register
0x814	GPUCLK_DIV	RW	0x0000_001F	32-bit	GPU Clock Divider Control register
0x818	-	RO	-	32-bit	Reserved, RAZ/WI
0x81C	-	RO	-	32-bit	Reserved, RAZ/WI
0x0820	GPUCORECLK_CTRL	RW	0x0000_0001	32-bit	GPU Clock Control register
0x0824	GPUCORECLK_DIV	RW	0x0000_001F	32-bit	GPU Clock Divider Control register

Offset	Name	Type	Reset	Width	Description
0x0830	GPUSTACKSCLK_CTRL	RW	0x0000_0001	32-bit	GPU Clock Control register
0x0834	GPUSTACKSCLK_DIV	RW	0x0000_001F	32-bit	GPU Clock Divider Control register
0x838 – 0x9FC	-	RO	-	-	Reserved, RAZ/WI
0xA00	CLKFORCE_STATUS	RO	-	32-bit	GPU Clock Force Status register
0xA04	CLKFORCE_SET	WO	-	32-bit	GPU Clock Force Set register
0xA08	CLKFORCE_CLR	WO	0x0	32-bit	GPU Clock Force Clear register
0xA0C – 0xFB8	-	RO	0x0	-	Reserved, RAZ/WI
0xFBC	CAP	RO	-	32-bit	GPU Capabilities register. Value dependent on chosen configuration.
0xFC0	PWR_CTRL_CONFIG	RO	-	32-bit	GPU Power Control Logic Configuration register. Value dependent on chosen configuration.
0xFD0	PID4	RO	0x44	32-bit	GPU Peripheral ID 4 register
0xFD4	PID5	RO	0x00	32-bit	GPU Peripheral ID 5 register
0xFD8	PID6	RO	0x00	32-bit	GPU Peripheral ID 6 register
0xFDC	PID7	RO	0x00	32-bit	GPU Peripheral ID 7 register
0xFE0	PID0	RO	0xB8	32-bit	GPU Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	32-bit	GPU Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	32-bit	GPU Peripheral ID 2 register
0xFEC	PID3	RO	0x00	32-bit	GPU Peripheral ID 3 register
0xFF0	CID0	RO	0x0D	32-bit	GPU Component ID 0 register
0xFF4	CID1	RO	0xF0	32-bit	GPU Component ID 1 register
0xFF8	CID2	RO	0x05	32-bit	GPU Component ID 2 register
0xFFC	CID3	RO	0xB1	32-bit	GPU Component ID 3 register

7.3.9.1 GPUCLK_CTRL, GPU Clock Control register

This register lets the user read the current source clock or change the source clock of GPUCLK, the GPU system clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

0x810

Type

0x0000_0001

Reset value

RW

Bit descriptions

Table 7-181: GPUCLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. 0x0 – No cycles 0x1 – 1 cycle ... 0xFF – 255 cycles This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK 0000_0100 – GPUPLLCLK Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK 0000_0100 – GPUPLLCLK Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.9.2 GPUCLK_DIV, GPU Clock Divider Control registers

This register lets the user read the current divider value or set a new value of GPUCLK, the GPU system clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0x814`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-182: GPUCLK_DIV bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, GPUCLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV_CUR + 1. For example, setting a value of 0 indicates a divider value of 1
[15:5]	-	Reserved
[4:0]	CLKDIV	GPUCLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example, GPUCLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV + 1. For example, setting a value of 0 indicates a divider value of 1

7.3.9.3 GPUCORECLK_CTRL, GPU Clock Control register

This register lets the user read the current source clock or change the source clock of GPUCORECLK, the GPU core group clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0x0820`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-183: GPUCORECLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK 0000_0100 – GPUPLLCLK Other values are Reserved

Bits	Name	Description
[7:0]	CLKSELECT	Selects the clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK 0000_0100 – GPUPLLCLK Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.9.4 GPUCORECLK_DIV, GPU Clock Divider Control registers

This register lets the user read the current divider value or set a new value of GPUCORECLK, the GPU core group clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0x0824`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-184: GPUCORECLK_DIV bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, GPUCLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV_CUR + 1. For example, setting a value of 0 indicates a divider value of 1
[15:5]	-	Reserved

Bits	Name	Description
[4:0]	CLKDIV	GPUCLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example, GPUCLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV + 1. For example, setting a value of 0 indicates a divider value of 1

7.3.9.5 GPUSTACKSCLK_CTRL, GPU Clock Control registers

This register lets the user read the current source clock or change the source clock of GPUSTACKSCLK, the GPU shader stack clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0x0830`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-185: GPUSTACKSCLK_CTRL bit descriptions

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. `0x0` – No cycles `0x1` – 1 cycle ... `0xFF` – 255 cycles This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[23:16]	-	Reserved

Bits	Name	Description
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK 0000_0100 – GPUPLLCLK Other values are Reserved
[7:0]	CLKSELECT	Selects the clock source 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK 0000_0100 – GPUPLLCLK Other values are Reserved. The result of writing one of the Reserved values into this field is **UNPREDICTABLE**.

7.3.9.6 GPUSTACKSCLK_DIV, GPU Clock Divider Control registers

This register lets the user read the current divider value or set a new value of GPUSTACKSCLK, the GPU shader stack clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0x0834`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-186: GPUSTACKSCLK_DIV bit descriptions

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, GPUCLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV_CUR + 1. For example, setting a value of 0 indicates a divider value of 1
[15:5]	-	Reserved
[4:0]	CLKDIV	GPUCLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example, GPUCLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV + 1. For example, setting a value of 0 indicates a divider value of 1

7.3.9.7 CLKFORCE_STATUS, GPU Clock Force Status register

The CLKFORCE_STATUS register records whether the dynamic clock gating associated with that clock is enabled (0) or disabled (1).

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xA00`

Type

RO

Bit descriptions

If a bit reads back as 1, then the associated dynamic clock gating associated with the clock is disabled, otherwise it is enabled.

Table 7-187: CLKFORCE_STATUS bit descriptions

Bits	Name	Description
[31:3]	-	Reserved
[3]	ELACLKFORCE	Clock force for ELA. This field is only available if CAP register bit 1 is set to 1. Otherwise, it is Reserved.
[2]	GPUSTACKSCLKFORCE	Clock force for GPUSTACKSCLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.

Bits	Name	Description
[2]	GPUCORECLKFORCE	Clock force for GPUCORECLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[1]	GPUCLKFORCE	Clock force for GPUCLK. This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[0]	-	Reserved

7.3.9.8 CLKFORCE_SET, GPU Clock Force SET register

Writing a 1 to a bit within the CLKFORCE_SET register disables any dynamic hardware clock gating for that respective clock, while writing 0 to a bit is ignored.

The CLFORCE_SET register share the same bit assignment as the CLKFORCE_STATUS register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xA04`

Type

WO

Bit descriptions

Table 7-188: CLKFORCE_SET bit descriptions

Bits	Name	Description
[31:3]	-	Reserved
[3]	ELACLKFORCE	Clock force for ELA. This field is only available if CAP register bit 1 is set to 1. Otherwise, it is Reserved.
[2]	GPUSTACKSCLKFORCE	Clock force for GPUSTACKSCLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[2]	GPUCORECLKFORCE	Clock force for GPUCORECLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[1]	GPUCLKFORCE	Clock force for GPUCLK. This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[0]	-	Reserved

7.3.9.9 CLKFORCE_CLR, GPU Clock Force Clear register

Writing a 1 to a bit within the CLKFORCE_CLR register enables the dynamic hardware clocking gating for that respective clock, while writing 0 to a bit is ignored.

The CLFORCE_CLR register shares the same bit assignment as the CLKFORCE_STATUS register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xA08`

Type

WO

Reset value

`0x0`

Bit descriptions

Table 7-189: CLKFORCE_CLR bit descriptions

Bits	Name	Description
[31:3]	-	Reserved
[3]	ELACLKFORCE	Clock force for ELA. This field is only available if CAP register bit 1 is set to 1. Otherwise, it is Reserved.
[2]	GPUSTACKSCLKFORCE	Clock force for GPUSTACKSCLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[2]	GPUCORECLKFORCE	Clock force for GPUCORECLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[1]	GPUCLKFORCE	Clock force for GPUCLK. This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[0]	-	Reserved

7.3.9.10 CAP, GPU Capabilities register

This register indicates various capabilities of the GPU block. For the type of capabilities indicated by the register, see the Bit descriptions section of the register definition.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFBC`

Type

RO

Bit descriptions

Table 7-190: CAP bit descriptions

Bits	Name	Description
[31:2]	-	Reserved
[1]	ELA support	0 – ELA is not supported 1 – ELA is supported All the fields in the GPU Power Control registers pertaining to ELA are *Read-As-Zero, Writes Ignored* (RAZ/WI) if this bit is set to 0.
[0]	GPUCLK Gating	0 – GPUCLK gating is not supported 1 – GPUCLK gating is supported.

7.3.9.11 PWR_CTRL_CONFIG, GPU Power Control Logic Configuration register

The power controller logical ID value is captured in this register. The value depends on the chosen configuration.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFC0`

Type

RO

Bit descriptions

Table 7-191: PWR_CTRL_CONFIG bit descriptions

Bits	Name	Description
[31:16]	-	POWER CONTROL LOGIC_ID. It is set to `0x41`.
[15:4]	-	Reserved
[3:0]	no_of_ppu	Defines the number of PPUs in the POWER CONTROL LOGIC. This value is set to `0x1` to indicate one *Power Policy Unit* (PPU).

7.3.9.12 PID4, GPU Peripheral ID4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFD0`

Type

RO

Reset value

`0x44`

Bit descriptions

Table 7-192: PID4 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).

Bits	Name	Description
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as `0x4`. This means that the POWER CONTROL LOGIC occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many `0x7F` continuation characters occur in the manufacturer's identity code. These bits read back as `0x4`.

7.3.9.13 PID5-7, GPU Peripheral ID5-7 register

The GPU Peripheral ID5-7 registers are *Read-As-Zero*, *Writes Ignored* (RAZ/WI).

7.3.9.14 PID0, GPU Peripheral ID0 register

The PID0 register contains the first eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFE0`

Type

RO

Reset value

`0xB8`

Bit descriptions

Table 7-193: PID0 bit description

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	part_number_0	These bits read back as `0xB8`

7.3.9.15 PID1, GPU Peripheral ID1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code and the second four bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFE4`

Type

RO

Reset value

`0xB0`

Bit descriptions

Table 7-194: PID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. See the JEP106, Standard Manufacturer's Identification Code.
[3:0]	part_number_1	These bits read back as `0x0`

7.3.9.16 PID2, GPU Peripheral ID2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

0xFE8

Type

RO

Reset value

0x0B

Bit descriptions

Table 7-195: PID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	Revision	Identifies the revision of the base POWER CONTROL LOGIC. For revision r0p0, this field is set to 0x0.
[3]	jedec_used	This indicates that the POWER CONTROL LOGIC uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.3.9.17 PID3, GPU Peripheral ID3 register

The PID3 register contains the manufacturer silicon revision number and a value that is incremented with each authorized customer modification.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

0xFEC

Type

RO

Reset value

0x00

Bit descriptions

Table 7-196: GPU Peripheral ID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)

Bits	Name	Description
[7:4]	RevAnd	The top-level RTL provides a 4-bit input, ECOREVNUM, that is normally tied LOW and provides a read value of `0x0`. When silicon is available, and if metal fixes are necessary, the manufacturer can modify the tie-offs to indicate a revision of the silicon.
[3:0]	mod_number	This is set to `0x0`

7.3.9.18 CID0, GPU Component ID 0 register

The CID_ID0 register contains segment 0 of the GPU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFF0`

Type

RO

Reset value

`0x0D`

Bit descriptions

Table 7-197: CID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_0	These bits read back as `0x0D`

7.3.9.19 CID1, GPU Component ID 1 register

The CID_ID1 register contains segment 1 of the GPU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-198: CID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_1	These bits read back as `0xF0`

7.3.9.20 CID2, GPU Component ID 2 register

The CID_ID2 register contains segment 2 of the GPU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-199: CID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_2	These bits read back as `0x05`

7.3.9.21 CID3, GPU Component ID 3 register

The CID_ID3 register contains segment 3 of the GPU component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU Power Control Logic (GPU PIK) registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-200: CID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_3	These bits read back as `0xB1`

7.3.10 SCP Power Control Logic (SCP PIK) registers

The SCP Power Control Logic registers provide access to parameters for the power control logic when configured for a *System Control Processor* (SCP).

The registers occupy 64KB which is split into 4KB blocks as shown in the following table.

Table 7-201: SCP PIK address offset

Offset	Description	Notes
0x0000	SCP PIK Control Registers	Clocks and SCP control registers
0x1000	SCP-PPU	Responsible for SCP logic
0x2000-0xFFFF	-	Reserved

The following table lists the SCP power control logic register summary.

Table 7-202: SCP Power Control Logic register summary

Offset	Name	Type	Reset	Width	Description
0x000- 0x00C	-	RW	0x0	-	Reserved, *Read-As-Zero, Writes Ignored* (RAZ/WI)
0x010	RESET_SYNDROME	RW	0x1	32-bit	Reset Syndrome register
0x014	WIC_CTRL	RW	0x0	32-bit	*Wakeup Interrupt Controller* (WIC) based Deep Sleep Control
0x018	WIC_STATUS	RO	0x0	32-bit	WIC based Deep Sleep Status
0x01C- 0x9FC	-	RW	0x0	-	Reserved, RAZ/WI
0xA00	CLKFORCE_STATUS	RO	-	32-bit	SCP Clock Force Status register
0xA04	CLKFORCE_SET	WO	-	32-bit	SCP Clock Force Set register
0xA08	CLKFORCE_CLR	WO	0x0	32-bit	SCP Clock Force Clear register
0xA0C	-	RO	0x0	-	Reserved, RAZ/WI
0xA10	PLL_STATUS0	RO	0x0	32-bit	PLL Status register 0
0xA14	PLL_STATUS1	RO	0x0	32-bit	PLL Status register 1
0xA18- 0xFBC	-	RO	0x0	-	Reserved, RAZ/WI
0xFC0	PWR_CTRL_CONFIG	RO	0x0070_0001	32-bit	Power control logic configuration register
0xFC4- 0xFCC	-	RW	0x0	-	Reserved, RAZ/WI
0xFD0	PID4	RO	0x44	32-bit	SCP Peripheral ID 4 register
0xFD4	PID5	RO	0x00	32-bit	SCP Peripheral ID 5 register
0xFD8	PID6	RO	0x00	32-bit	SCP Peripheral ID 6 register
0xFDC	PID7	RO	0x00	32-bit	SCP Peripheral ID 7 register
0xFE0	PID0	RO	0xB8	32-bit	SCP Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	32-bit	SCP Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	32-bit	SCP Peripheral ID 2 register
0xFEC	PID3	RO	0x00	32-bit	SCP Peripheral ID 3 register
0xFF0	CID0	RO	0x0D	32-bit	SCP Component ID 0 register
0xFF4	CID1	RO	0xF0	32-bit	SCP Component ID 1 register
0xFF8	CID2	RO	0x05	32-bit	SCP Component ID 2 register
0xFFC	CID3	RO	0xB1	32-bit	SCP Component ID 3 register

7.3.10.1 RESET_SYNDROME, Reset Syndrome register

The RESET_SYNDROME register captures the cause for the last reset.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0x010`

Type

RW

Reset value

`0x1`

Bit descriptions

Table 7-203: RESET_SYNDROME bit descriptions

Bits	Name	Description
[31:5]	-	Reserved
[4]	SCPM3LOCKUP	Indicates that before last reset the *System Control Processor* (SCP) Cortex-M3 was in the Lockup state. SCP firmware should write a 0 to this location to clear this bit post reset.
[3]	SYSRESETREQ	Last reset was caused by SYSRESETREQ bit in AIRCR of SCP Cortex-M3. SCP firmware should write a 0 to this location to clear this bit post reset.
[2]	WDOGRESET_SYS	Last reset was caused by System Trusted Watchdog. SCP firmware should write a 0 to this location to clear this bit post reset.
[1]	WDOGRESET_SCP	Last reset was caused by SCP Watchdog. SCP firmware should write a 0 to this location to clear this bit post reset.
[0]	PORESETn	Last reset was caused by PORESETn input. SCP firmware should write a 0 to this location to clear this bit post reset.

7.3.10.2 WIC_CTRL, WIC based Deep Sleep Control register

This register controls if the next SCP sleep is a WIC sleep or not.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

0x014

Type

RW

Reset value

0x0

Bit descriptions

Table 7-204: WIC_CTRL bit descriptions

Bits	Name	Description
[31:1]	-	Reserved
[0]	WIC_EN	Enable *Wakeup Interrupt Controller* (WIC) based Deep Sleep 0 – Request that the next Deep Sleep be WIC-based 1 – Request that the next Deep Sleep not be WIC-based

7.3.10.3 WIC_STATUS, WIC based Deep Sleep Status register

This register indicates if SCP WIC sleep is enabled or not.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

0x018

Type

RO

Reset value

0x0

Bit descriptions

Table 7-205: WIC_STATUS bit description

Bits	Name	Description
[31:1]	-	Reserved
[0]	WICEN_STATUS	Status of *Wakeup Interrupt Controller* (WIC) based Deep Sleep 0 – WIC-based Deep Sleep Enabled 1 – WIC-based Deep Sleep Disabled

7.3.10.4 CLKFORCE_STATUS, Clock Force Status register

The CLKFORCE_STATUS register records whether the dynamic clock gating associated with that clock is enabled (0) or disabled (1).

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xA00`

Type

RO

Bit descriptions

If a bit reads back as 1, then the associated dynamic clock gating associated with the clock is disabled, otherwise it is enabled.

Table 7-206: CLKFORCE_STATUS bit descriptions

Bits	Name	Description
[31:2]	-	Reserved
[1]	SCPCLKFORCE	Force the internal SCPCLK clock on
[0]	REFCLKFORCE	Force the REFCLK clock on

7.3.10.5 CLKFORCE_SET, SCP Clock Force SET register

This register provides the ability to disable dynamic clock gating on the respective clock.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xA04`

Type

WO

Bit descriptions

Writing a 1 to a bit within the CLKFORCE_SET register disabled any dynamic hardware clock gating for that respective clock, while writing 0 to a bit is ignored.

Table 7-207: CLKFORCE_SET bit descriptions

Bits	Name	Description
[31:2]	-	Reserved
[1]	SCPCLKFORCE	Force the internal SCPCLK clock on
[0]	REFCLKFORCE	Force the REFCLK clock on

7.3.10.6 CLKFORCE_CLR, Clock Force Clear register

Writing a 1 to a bit within the CLKFORCE_CLR register enables the dynamic hardware clocking gating for that respective clock, while writing 0 to a bit is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xA08`

Type

WO

Reset value

0x0

Bit descriptions

Table 7-208: CLKFORCE_CLR bit descriptions

Bits	Name	Description
[31:2]	-	Reserved
[1]	SCPCLKFORCE	Force the internal SCPCLK clock on
[0]	REFCLKFORCE	Force the REFCLK clock on

7.3.10.7 PLL_STATUS0, PLL Status 0 register

This register indicates the lock status of the corresponding PLL.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic register

Address offset

0xA10

Type

RO

Reset value

0x0

Bit descriptions

Table 7-209: PLL_STATUS0 bit descriptions

Bits	Name	Description
[31:8]	-	IMPLEMENTATION DEFINED
[7]	NPUPLLLOCK_ST	Status of the NPU *Phase-Locked Loop* (PLL) Lock
[6]	DISPLAYPLLLOCK_ST	Status of the Display PLL Lock
[5]	SYSPLLLLOCK_ST	Status of the System PLL Lock
[4]	VIDEOPLLLOCK_ST	Status of the Video PLL Lock
[3]	GPUPLLLOCK_ST	Status of the GPU PLL Lock

Bits	Name	Description
[2:1]	-	Reserved
[0]	REFCLK_ST	Status of REFCLK

7.3.10.8 PLL_STATUS1, PLL Status 1 register

This register indicates the lock status of the corresponding PLL.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xA14`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-210: PLL_STATUS1 bit descriptions

Bits	Name	Description
[31:7]	-	IMPLEMENTATION DEFINED
[15]	CPU1PLL7 LOCK _ST	Status of CPU1 PLL7
[14]	CPU1PLL6 LOCK _ST	Status of CPU1 PLL6
[13]	CPU1PLL5 LOCK _ST	Status of CPU1 PLL5
[12]	CPU1PLL4 LOCK _ST	Status of CPU1 PLL4
[11]	CPU1PLL3 LOCK _ST	Status of CPU1 PLL3
[10]	CPU1PLL2 LOCK _ST	Status of CPU1 PLL2
[9]	CPU1PLL1 LOCK _ST	Status of CPU1 PLL1
[8]	CPU1PLL0 LOCK _ST	Status of CPU1 PLL0
[7]	CPU0PLL7 LOCK _ST	Status of CPU0 PLL7
[6]	CPU0PLL6 LOCK _ST	Status of CPU0 PLL6
[5]	CPU0PLL5 LOCK _ST	Status of CPU0 PLL5
[4]	CPU0PLL4 LOCK _ST	Status of CPU0 PLL4
[3]	CPU0PLL3 LOCK _ST	Status of CPU0 PLL3

Bits	Name	Description
[2]	CPU0PLL2 LOCK _ST	Status of CPU0 PLL2
[1]	CPU0PLL1LOCK _ST	Status of CPU0 PLL1
[0]	CPU0PLL0LOCK_ST	Status of CPU0 PLL0

7.3.10.9 PWR_CTRL_CONFIG, SCP Power Control Logic Configuration register

The power controller logical ID value is captured in this register. The value depends on the chosen configuration.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xFC0`

Type

RO

Reset value

`0x0070_0001`

Bit descriptions

Table 7-211: PWR_CTRL_CONFIG bit descriptions

Bits	Name	Description
[31:16]	-	POWER CONTROL LOGIC_ID. This field is set to `0x0073`.
[15:4]	-	Reserved
[3:0]	no_of_ppu	Defines the number of *Power Policy Units* (PPUs) in the POWER CONTROL LOGIC. This value is set to `0x1` to indicate one PPU.

7.3.10.10 PID4, SCP Peripheral ID 4 register

The PID4 register contains information about the number of address blocks that the logic occupies and the JEDEC JEP106 configuration code for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xFD0`

Type

RO

Reset value

`0x44`

Bit descriptions

Table 7-212: PID4 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as `0x4`. This means that the POWER CONTROL LOGIC occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many `0x7F` continuation characters occur in the manufacturer's identity code. These bits read back as `0x4`.

7.3.10.11 PID5-7, SCP Peripheral ID 5-7 register

The SCP Peripheral ID 5-7 registers are Reserved, *Read-As-Zero*, *Writes Ignored* (RAZ/WI).

7.3.10.12 PID0, SCP Peripheral ID 0 register

The PID0 register contains the first eight bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xFE0`

Type

RO

Reset value

0xB8

Bit descriptions

Table 7-213: PID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	part_number_0	These bits read back as 0xB8

7.3.10.13 PID1, SCP Peripheral ID 1 register

The PID1 register contains the first four bits of the JEDEC JEP106 identity code and the second four bits of the identifier for the peripheral.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

0xFE4

Type

RO

Reset value

0xB0

Bit descriptions

Table 7-214: PID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. See the JEP106, Standard Manufacturer's Identification Code. These bits read back as 0xB because Arm is the peripheral designer.
[3:0]	part_number_1	These bits read back as 0x0.

7.3.10.14 PID2, SCP Peripheral ID 2 register

The PID2 register specifies whether the JEDEC JEP106 identification scheme is in use, and contains parts of the peripheral JEP106 designer code and block version.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xFE8`

Type

RO

Reset value

`0x0B`

Bit descriptions

Table 7-215: PID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).
[7:4]	Revision	Identifies the revision of the base POWER CONTROL LOGIC. For revision r0p0, this field is set to `0x0`.
[3]	jedec_used	This indicates that the POWER CONTROL LOGIC uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as `0x1`
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code. These bits read back as `0x3` because Arm is the peripheral designer.

7.3.10.15 PID3, SCP Peripheral ID 3 register

The PID3 register contains the manufacturer silicon revision number and a value that is incremented with each authorized customer modification.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

0xFEC

Type

RO

Reset value

0x00

Bit descriptions

Table 7-216: PID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:4]	RevAnd	The top level RTL provides a 4-bit input, ECOREVNUM, which is normally tied LOW and provides a read value of 0x0. When silicon is available, and if metal fixes are necessary, the manufacturer can modify the tie-offs to indicate a revision of the silicon.
[3:0]	mod_number	This is set to 0x0

7.3.10.16 CID0, SCP Component ID 0 register

The CID0 register contains segment 0 of the debug chain power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

0xFF0

Type

RO

Reset value

0x0D

Bit descriptions

Table 7-217: CID0 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_0	These bits read back as `0x0D`

7.3.10.17 CID1, SCP Component ID 1 register

The CID1 register contains segment 1 of the debug chain power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-218: CID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_1	These bits read back as `0xF0`

7.3.10.18 CID2, SCP Component ID 2 register

The CID2 register contains segment 2 of the system power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-219: CID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_2	These bits read back as `0x05`

7.3.10.19 CID3, SCP Component ID 3 register

The CID3 register contains segment 3 of the system power control component class identifier.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP Power Control Logic (SCP PIK) registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-220: CID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ)
[7:0]	comp_id_3	These bits read back as `0xB1`

Appendix A Revisions

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

Table A-1: Issue 0000-01

Change	Location
First release	-