Arm® Total Compute 2021 Reference Design

Software Developer Guide

Non-Confidential

Issue 01

107591_0000_01_en

Arm® Total Compute 2021 Reference Design
Software Developer Guide

Release information

Document history

Issue	Date	Confidentiality	Change
0000-01	27 October 2022	Non-Confidential	First release

Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED “AS IS”. ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws

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 Intended audience

The Software Developer Guide (SDG) is intended to assist software developers working with the related Fixed Virtual Platform (FVP) model and Open Source Software stack.

1.2 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	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.

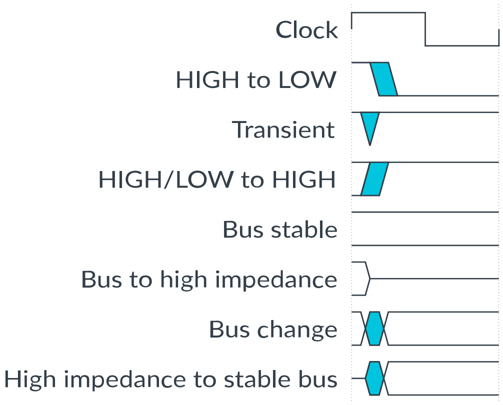
Convention	Use
⚠ Danger	Requirements for the system. Not following these requirements will result in system failure or damage.
✎ Note	An important piece of information that needs your attention.
💡 Tip	A useful tip that might make it easier, better or faster to perform a task.
📌 Remember	A reminder of something important that relates to the information you are reading.

Timing diagrams

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

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

Figure 1-1: Key to timing diagram conventions



Signals

The signal conventions are:

Signal level

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

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

Lowercase n

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

1.3 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 Total Compute solution – Arm Developer	–	Non-Confidential
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™ 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®-A510 Core Cryptographic Extension Technical Reference Manual	101606	Non-Confidential
Arm® Cortex®-A510 Core Technical Reference Manual	101604	Non-Confidential
Arm® Cortex®-A715 Core Cryptographic Extension Technical Reference Manual	101592	Non-Confidential
Arm® Cortex®-A715 Core Technical Reference Manual	101590	Non-Confidential
Arm® Cortex®-M3 Processor Technical Reference Manual	100165	Non-Confidential
Arm® Cortex®-X3 Core Cryptographic Extension Technical Reference Manual	101602	Non-Confidential
Arm® Cortex®-X3 Core Technical Reference Manual	101593	Non-Confidential
Arm® DynamIQ™ Shared Unit-110 Technical Reference Manual	101381	Non-Confidential
CoreSight™ Components Technical Reference Manual	DDI 0314	Non-Confidential
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 0022H	Non-Confidential
AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces	IHI 0068	Non-Confidential
Arm® Architecture Reference Manual for A-profile architecture	DDI 0487	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Base System Architecture	DEN 0094	Non-Confidential
Arm® CoreSight Base System Architecture 1.0, Arm Platform Design Document	DEN 0068	Non-Confidential
Arm® Debug Interface Architecture Specification ADIv5.0 to ADIv5.2	IHI 0031	Non-Confidential
Arm® Debug Interface Architecture Specification ADIv6.0	IHI 0074C	Non-Confidential
Arm® Generic Interrupt Controller Architecture Specification, GIC architecture version 3 and version 4	IHI 0069G	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 3.2	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

Note

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

2. Overview of Total Compute 2021 Reference Design

Arm® Total Compute 2021 Reference Design (RD-TC21) describes and models the design choices for recommended configurations of typical Arm-based *Compute SubSystem* (CSS).

The RD-TC21 subsystem addresses the expected requirements in the Premium Mobile market.

Features

RD-TC21 supports Arm®v8.5-A and Arm®v9.0-A architecture extensions, and provides key features built around the following 2021-generation IP:

- A single processor cluster, up to a maximum of 12 cores in a combination of the following cores with DynamIQ™ Shared Unit-110:
  - Cortex®-X3 Core
  - Cortex®-A715 Core
  - Cortex®-A510 Core
- Mali™-G715 GPU
  - Supports ray tracing
  - Supports double arithmetic capabilities compared to the previous generation GPUs

See Hardware and topology for more information on Arm IP, system architecture, and key features of each block.

The RD-TC21 *Fixed Virtual Platform* (FVP) is a functional model intended for software development on the RD-TC21 subsystem configurations. See FVP for more information on supported reference configuration.

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

Total Compute 2021 Reference Design (RD-TC21) includes deliverables to explore the design from a software perspective.

- A Software Developer Guide describes a high-level overview of RD-TC21, including the reference system architecture from which design has been derived, the software stack, and the FVP.
- *Fixed Virtual Platform* (FVP) provides a software model of the RD-TC21 subsystem to explore from a software perspective.
- Software provides a starting point to modify, extend, and develop the software stack for a SoC derived from the RD-TC21.

2.2 Compliance

The Total Compute 2021 Reference Design complies 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*
- *AMBA® 5 CHI Architecture Specification*
- *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 3.2*
- *Trusted Base System Architecture, Client (4TH Edition)*

3. Hardware and topology

The architecture of Total Compute 2021 Reference Design supports the subsystem design of premium mobile market segments.

3.1 IP in RD-TC21

The IP and components in Total Compute 2021 Reference Design (RD-TC21) provide a reference design that you can use to create custom SoC.

The RD-TC21 contains the following IP:

- Arm® Cortex®-A715 Core
- Arm® Cortex®-X3 Core
- Arm® Cortex®-A510 Core (r0)
- Arm® DynamIQ™ Shared Unit-110
- Arm® Mali™-G715 GPU
- Arm® CoreLink™ CI-700 Coherent Interconnect (r1)
- Arm® CoreLink™ NI-700 Network-on-Chip Interconnect (r1)
- Arm® CoreLink™ GIC-700 Generic Interrupt Controller (r0)
- Arm® CoreLink™ MMU-700 System Memory Management Unit (r0)
- *System Control Processor* (SCP) based on the Arm® Cortex®-M3 processor
- Arm® CoreLink™ NIC-450 Network Interconnect
- Arm® CoreLink™ PCK-600 Power Control Kit
- Arm® CoreSight™ System-on-Chip SoC-600
- On-Chip ROM, RAM, and other peripherals
- Clock generators with support for dynamic clock gating

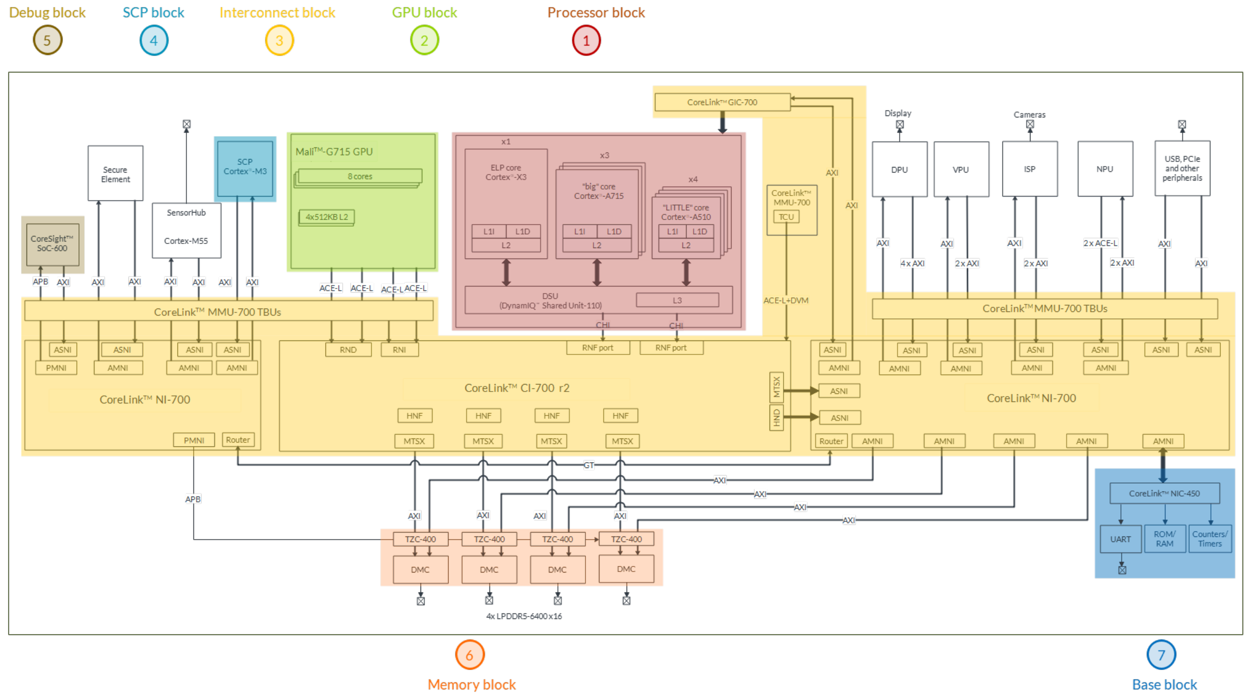
3.2 System architecture

The Total Compute 2021 Reference Design (RD-TC21) is partitioned into functional blocks that are a combination of major IP and the supporting logic around them.

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

The following figure shows a high-level view of architecture of the RD-TC21.

Figure 3-1: RD-TC21 system architecture



The functional blocks in RD-TC21 are:

Processor block, see Processor block

The Processor block contains the following new generation cores based on Arm®v8.5-A and Arm®v9.0-A architectures:

- Cortex®-A715 Core
- Cortex®-X3 Core
- Cortex®-A510 Core
- DynamIQ™ Shared Unit-110

GPU block, see GPU block

The GPU block contains:

- Mali™-G715 GPU with configurable number of shader cores
- Dual-voltage domains support and power-gated regions

Interconnect block, see Interconnect block

The Interconnect block contains:

- CoreLink™ CI-700 for the premium mobile system
- Configurable CoreLink™ NI-700 for non-coherent connecting to various master and slave extension interfaces

- *GIC distributor* (GIC-D) and *Interrupt Translation Service* (ITS) logic, which CoreLink™ GIC-700 implements
- CoreLink™ MMU-700 with *Translation Control Unit* (TCU) and *Translation Buffer Unit* (TBU) logic

SCP block, see SCP block

The SCP block contains:

- Cortex®-M3 based *System Control Processor* (SCP). It also connects to all power control logic within each block that has the relevant power control logic.
- MHUv2.0 for processor-to-processor message communication support

Memory block, see Memory block

The Memory block contains a memory controller that implements data path from the cache-coherent interconnect. The Memory block wraps the dynamic memory controller with the necessary glue-logic to integrate with the rest of the subsystem.

It contains the inline external TrustZone controller TZC-400, an Address translation function, power management, clock and reset generation, and control logic.

Base block, see Base block

The Base block contains system peripherals including generic timers, watchdog timers, scratch RAMs and ROMs. The Base block also includes a CoreLink™ NIC-450 interconnect, which the peripherals connect to.

3.3 Processor block

The Total Compute 2021 Reference Design (RD-TC21) has a single processor block. The Processor block contains the new generation cores based on Arm®v8.5-A and Arm®v9.0-A architectures. The cores are wrapped in the *DynamIQ™ Shared Unit* (DSU) and other support logic such as debug logic, and reset and clock generation.

The cores in the Processor block are also referred to as *Application Processor* (AP).

The following three tiers of cores can be used in the Processor block for optimizing both performance and power:

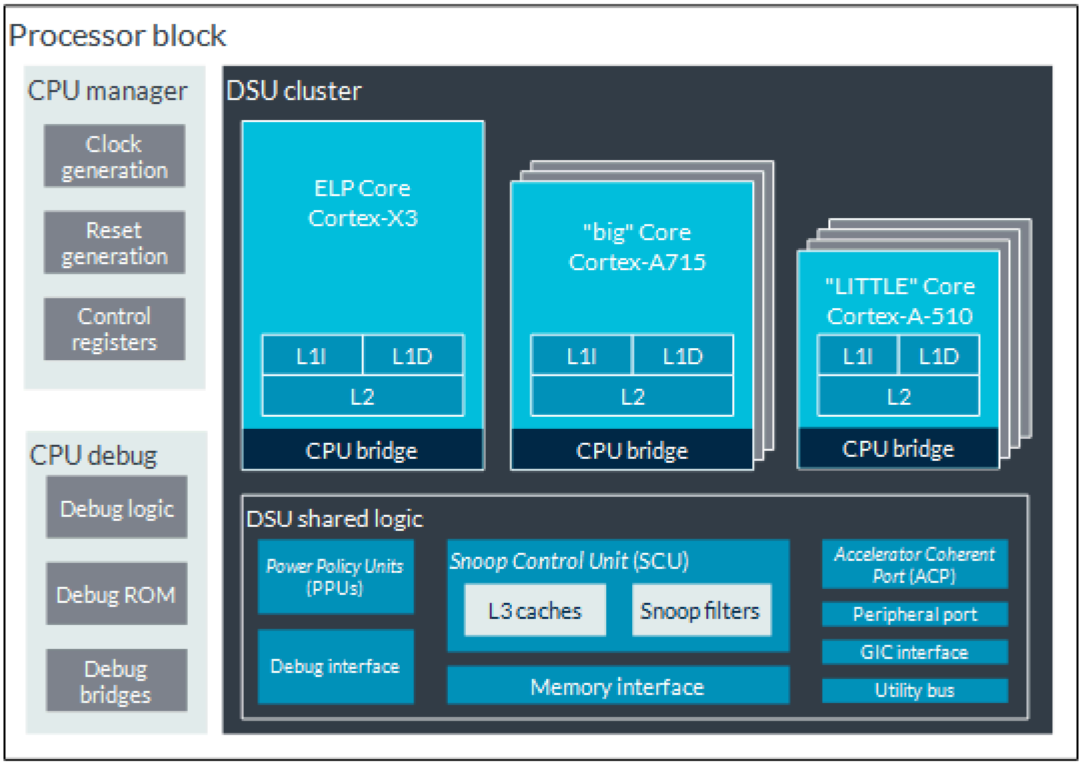
- ELP core: Cortex®-X3 Core
- “big” core: Cortex®-A715 Core
- “LITTLE” core: Cortex®-A510 Core

These processor cores share the following listed high-level Armv8.5-A and Armv9.0-A features:

- Memory Tagging
- *Scalable Vector Extensions* (SVE). For more information about SVE and SVE2, see SVE - Arm Developer page.

The following figure shows a high-level representation of the Processor block.

Figure 3-2: Block diagram of Processor block



For information about clock and reset generation, and power management of this block, see the corresponding subsections in Functional description.

3.3.1 Cortex®-X3 core

The Cortex®-X3 Core can be used in a standalone DynamIQ™ configuration, which is a homogenous cluster of one to four Cortex-X3 cores. It might also be used as the high performance core in a heterogenous cluster.

However, regardless of the cluster configuration, the Cortex-X3 core always has the same features.

For more information, see the following documentation:

- *Arm® Cortex®-X3 Core Technical Reference Manual*
- *Arm® Cortex®-X3 Core Cryptographic Extension Technical Reference Manual*

Core features

The Cortex-X3 core provides the following core features:

- Implementation of the Arm®v9.0-A A64 instruction set

- AArch64 Execution state at all Exception levels, EL0 to EL3
- *Memory Management Unit* (MMU)
- 40-bit *Physical Address* (PA) and 48-bit *Virtual Address* (VA)
- *Generic Interrupt Controller* (GIC) CPU interface to connect to an external interrupt distributor
- Generic Timers interface that supports 64-bit count input from an external system counter
- Implementation of the *Reliability, Availability, and Serviceability* (RAS) Extension
- Implementation of the *Scalable Vector Extension* (SVE) with a 128-bit vector length and Scalable Vector Extension 2 (SVE2)
- Integrated execution unit with Advanced *Single Instruction Multiple Data* (SIMD) and floating point support
- Implementation of *Memory Tagging Extension* (MTE) including asymmetric MTE support
- Support for the optional Cryptographic Extension, which is licensed separately
- *Activity Monitoring Unit* (AMU)

Cache features

The Cortex-X3 core provides the following cache features:

- Separate L1 data and instruction caches
- Private, unified data and instruction L2 cache
- Error protection on L1 instruction and data caches, L2 cache, and *MMU Translation Cache* (MMU TC) with parity or *Error Correcting Code* (ECC) allowing *Single Error Correction and Double Error Detection* (SECDED).
- Support for *Memory System Resource Partitioning and Monitoring* (MPAM)

Debug features

The Cortex-X3 core provides the following debug features:

- Armv9.0-A debug logic
- *Performance Monitoring Unit* (PMU)
- *Embedded Trace Macrocell* (ETM) with support for *Embedded Trace Extension* (ETE)
- *Trace Buffer Extension* (TRBE)
- Support for *Statistical Profiling Extension* (SPE)

3.3.2 Cortex®-A715 core

You can use the Cortex®-A715 Core in a standalone DynamIQ™ configuration, that is, in a homogenous cluster of one to four Cortex-A715 cores. You can also use the Cortex-A715 core as the balanced-performance core in a heterogenous cluster.

However, regardless of the cluster configuration, the Cortex-A715 core always has the same features as described in the following lists.

For more information, see the following documentation:

- *Arm® Cortex®-A715 Core Technical Reference Manual*
- *Arm® Cortex®-A715 Core Cryptographic Extension Technical Reference Manual*

Core features

The Cortex-A715 core provides the following core features:

- Implementation of the Arm®v9.0-A A64 instruction set
- AArch64 Execution state at all Exception levels, EL0 to EL3
- *Memory Management Unit* (MMU)
- 40-bit *Physical Address* (PA) and 48-bit *Virtual Address* (VA)
- *Generic Interrupt Controller* (GIC) CPU interface to connect to an external interrupt distributor
- Generic Timers interface that supports 64-bit count input from an external system counter
- Implementation of the *Reliability, Availability, and Serviceability* (RAS) extension
- Implementation of the *Scalable Vector Extension* (SVE) with a 128-bit vector length and *Scalable Vector Extension 2* (SVE2)
- Integrated execution unit with Advanced *Single Instruction Multiple Data* (SIMD) and floating point support
- Support for the optional Cryptographic Extension, which is licensed separately
- *Activity Monitoring Unit* (AMU)

Cache features

The Cortex-A715 core provides the following cache features:

- Separate L1 data and instruction caches
- Private, unified data and instruction L2 cache
- Optional error protection with parity or *Error Correcting Code* (ECC) allowing *Single Error Correction and Double Error Detection* (SECDED) on L1 instruction and data caches, L2 cache, and L2 *Translation Lookaside Buffer* (TLB)
- Support for *Memory System Resource Partitioning and Monitoring* (MPAM)

Debug features

The Cortex-A715 core provides the following debug features:

- Armv9.0-A debug logic
- *Performance Monitoring Unit* (PMU)
- *Embedded Trace Macrocell* (ETM) with support for *Embedded Trace Extension* (ETE)
- *Trace Buffer Extension* (TRBE)
- Optional implementation of the *Statistical Profiling Extension* (SPE)

3.3.3 Cortex®-A510 core

The Cortex®-A510 Core might be used in standalone DynamIQ™ configurations where a homogenous DynamIQ™ Shared Unit-110 cluster includes one to eight Cortex-A510 cores.

The Cortex-A510 core might also be used as a high efficiency core or a high-performance core in a heterogenous DSU-110 cluster.

However, regardless of the cluster configuration, the Cortex-A510 core always has the same features.

For more information, see the following documentation:

- *Arm® Cortex®-A510 Core Technical Reference Manual*
- *Arm® Cortex®-A510 Core Cryptographic Extension Technical Reference Manual*

Core features

The Cortex-A510 provides the following core features:

- Implementation of the Arm®v9.0-A A64 instruction set
- AArch64 Execution state at all Exception levels, EL0 to EL3
- Separate L1 data and instruction side memory systems with a *Memory Management Unit* (MMU)
- In-order pipeline with direct and indirect branch prediction
- *Generic Interrupt Controller* (GIC) CPU interface to connect to an external interrupt distributor
- Generic Timer interface that supports a 64-bit count input from an external system counter
- Implementation of the *Reliability, Availability, and Serviceability* (RAS) Extension
- *Scalable Vector Extension* (SVE) and SVE2 SIMD instruction set, offering Advanced *Single Instruction Multiple Data* (SIMD) and floating-point architecture support
- Support for the optional Cryptographic Extension, which is licensed separately
- *Activity Monitoring Unit* (AMU)

Cache features

The Cortex-A510 provides the following cache features:

- Separate L1 data and instruction caches
- Optional unified L2 cache
- L1 and L2 cache protection with *Error Correcting Code* (ECC) or parity
- Support for *Memory System Resource Partitioning and Monitoring* (MPAM)

Debug features

The Cortex-A510 provides the following debug features:

- Armv9.0-A debug logic

- *Performance Monitoring Unit* (PMU)
- *Embedded Trace Macrocell* (ETM) with support for *Embedded Trace Extension* (ETE)
- *Trace Buffer Extension* (TRBE)

3.3.4 DynamIQ™ Shared Unit-110 cluster

The DynamIQ™ Shared Unit-110 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 the DSU-110 cluster.

Also, all the external interfaces to the *System on Chip* (SoC) are provided through the DSU-110.

For more information, see *Arm® DynamIQ™ Shared Unit-110 Technical Reference Manual*.

Cache features

The DSU-110 provides the following cache features:

- Optional unified 16-way set-associative L3 cache, configurable from 256KB to 16MB
- 64-byte cache lines
- L3 cache slice support, for improved bandwidth and cache RAM layout, up to eight slices supported
- L3 cache powerdown based either on cache slices or cache ways
- Cache partitioning support, compliant with *Memory System Resource Partitioning and Monitoring* (MPAM) architecture
- *Error Correcting Code* (ECC) protection on L3 cache RAM instances
- L3 cache system can be clocked at a rate synchronous to the external system interconnect or at integer multiples

Coherency and snoop control features

The DSU-110 provides the following coherency and snoop control features:

- *Snoop Control Unit* (SCU) maintains coherency and consistency in the memory system internal to the cluster, and (optionally) external to the cluster.
- SCU includes a set of snoop filters, automatically sized, one for each cache slice

Cluster features

The DSU-110 has the following cluster features:

- Support for Arm®v9.0-A architecture cores
- Support for up to four types of core, and a maximum of 12 cores in the cluster
- *Power Policy Units* (PPUs) providing autonomous power management of the L3 cache and the cores
- Support for cores running independently at different frequencies and voltages known as *Dynamic Voltage Frequency Scaling* (DVFS). For cores in a complex, DVFS is only possible for the whole complex not for individual cores.

- The DSU-110 has an internal transport mechanism that is responsible for all communication between components in the design. The topology of the transport is defined by the number of cores and number of L3 cache slices.

Interface features

The DSU-110 provides the following interface features:

- Optional AMBA 5 CHI Issue E 256-bit coherent master bus interface that supports up to four CHI bus master ports.
- Optional AMBA AXI5 Issue H 256-bit non-coherent master interface that supports up to four AXI bus master ports.
- Optional 128-bit or 256-bit wide I/O-coherent *Accelerator Coherency Port* (ACP) based on AMBA ACE5-Lite.
- AMBA AXI5 Utility bus providing programming interface to PPUs, and other system components.
- Optional Peripheral Port interface that is implemented as either an AXI 64-bit wide port, AXI 256-bit wide port, or CHI 256-bit wide port.
- Simplified system integration for interfaces such as debug and trace which are already in the correct clock domain at the output of the cluster.

Debug and trace features

The DSU-110 provides the following debug and trace features:

- Debug-over-powerdown support
- CoreSight™ SoC-600 support for the *Embedded Trace Macrocell* (ETM) and *Cross Trigger Interface* (CTI) for each core

3.3.4.1 DynamIQ™ Shared Unit-110 interrupts

All *Peripheral Port Interface* (PPI) CPU interrupts are routed through a shared Generic Interrupt Controller, CoreLink™ GIC-700. All other error or fault interrupts are routed to *System Control Processor* (SCP).

Error or fault interrupts from the DSU-110 cluster

The status of these error or fault interrupts are captured in the status register in the Processor block manager. The combined interrupt output is sent to SCP.

Table 3-1: Error or fault interrupts from the DSU-110 cluster

Signal	Direction	Description
nCLUSTERFAULTIRQ	Output	Fault indicator for a detected 1 or 2 bit ECC/Parity error in the L3 or snoop filter RAMs.
nCLUSTERERRIRQ	Output	Error indicator for an ECC error that causes potential data corruption or loss of coherency in the L3 or snoop filter RAMs or AXI/CHI write transactions with a write response error condition.
nCLUSTERCRITIRQ	Output	Error indicator for a critical error, that is, an uncorrectable snoop filter error. The DSU-110 generates an interrupt, nCLUSTERCRITIRQ, to notify software that data has potentially lost.
nMPAMIRQ	Output	Error indicator from MPAM programming.

Error or fault interrupts from each core with in the DSU-110 cluster

The status of these error or fault interrupts are captured in the status register in the Processor block manager. The combined interrupt output is sent to SCP.

Table 3-2: Error or fault interrupts from each core with in the DSU-110 cluster

Signal	Direction	Description
nCOREFAULTIRQ[CN:0]	Output	Fault indicator for a detected 1 or 2-bit ECC/Parity error in the RAMs. Bits CN down to 0 are for the L1 and L2 RAMs in each core.
nCOMPLEXFAULTIRQ[CX:0]	Output	Fault indicator for a detected 1 or 2-bit ECC/Parity error in the RAMs. Bits CX down to 0 are for the shared RAMs in each complex. Note: This is applicable only for Cortex®-A510 Core.
nCOREERRIRQ[CN:0]	Output	Error indicator for an ECC error that causes potential data corruption or loss of coherency. Bits CN down to 0 are for the L1 and L2 RAMs in each core.
nCOMPLEXERRIRQ[CX:0]	Output	Error indicator for an ECC error that causes potential data corruption or loss of coherency. Bits CX down to 0 are for the shared RAMs in each complex. Note: This is applicable only for Cortex®-A510 Core.

3.4 GPU block

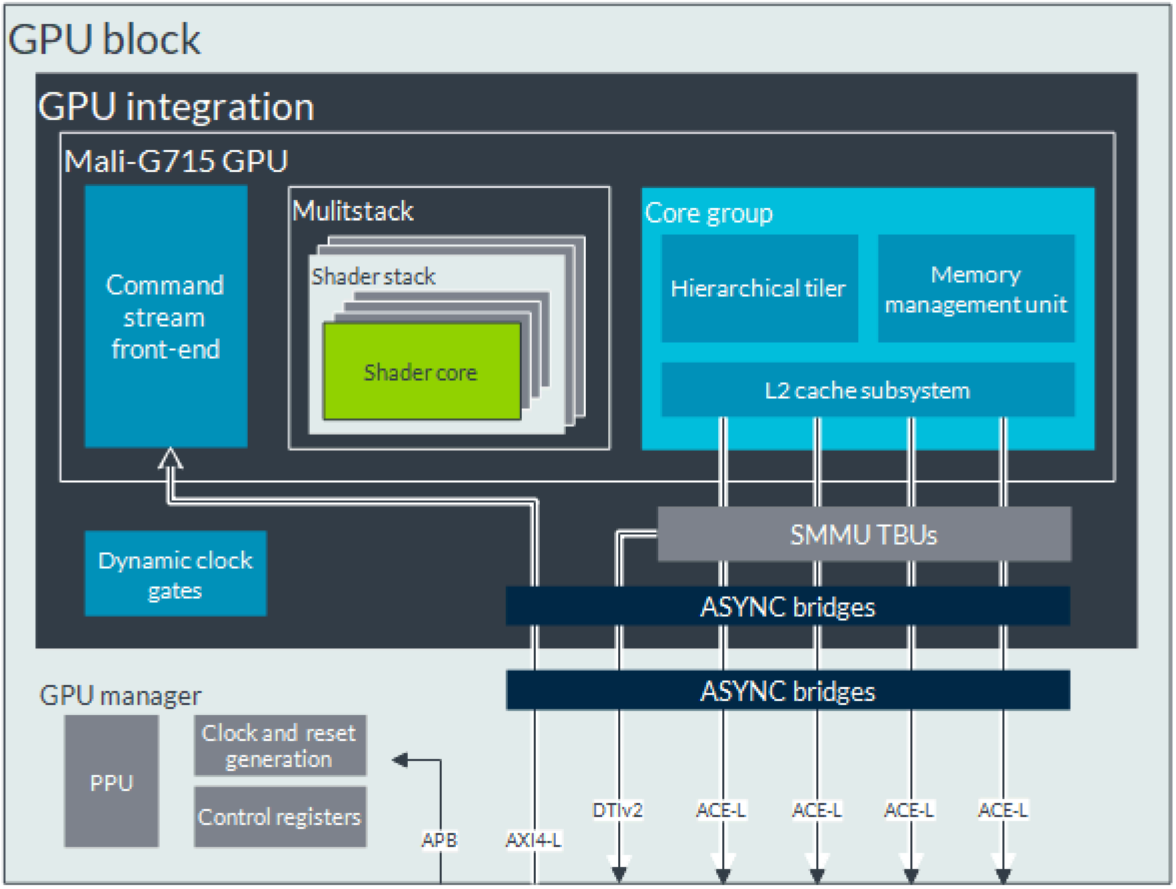
The GPU block is implemented by Mali™-G715 GPU with a configurable number of shader cores.

Mali™-G715 GPU supports the following standards:

- OpenGL ES 1.1, 2.0, 3.x
- Vulkan 1.2
- OpenCL 1.x and 2.x full profile

The following figure shows a high-level representation of the GPU block.

Figure 3-3: Block diagram of GPU block



In the Block diagram of GPU block, the Mali-G715 GPU is the main component. It supports configurable shader cores for different *Power, Performance, and Area* (PPA) targets. The Mali-G715 GPU also supports Arm® graphic compression technology which significantly reduces its bandwidth to main memory. This reduces power consumption and improves overall system performance.

Features

The Mali-G715 GPU provides the following features:

- Support of multiple clock and voltage domains, allowing you to maximize performance within a given power budget
- Programmable architecture
- API feature set with support for shader-based and fixed-function graphics APIs
- Anti-aliasing capabilities
- Effective core for *General Purpose computing on GPU* (GPGPU) applications
- High memory bandwidth and low power consumption for 3D graphics content
- Scalability for products from smartphones to high-end mobile computing

- Performance-leading 3D graphics
- AMBA 4 AXI-Lite slave interface for GPU configuration
- 128-bit or 256-bit AMBA 4 ACE master interface for external memory access

Note

The Mali-G715 GPU does not support full coherency.

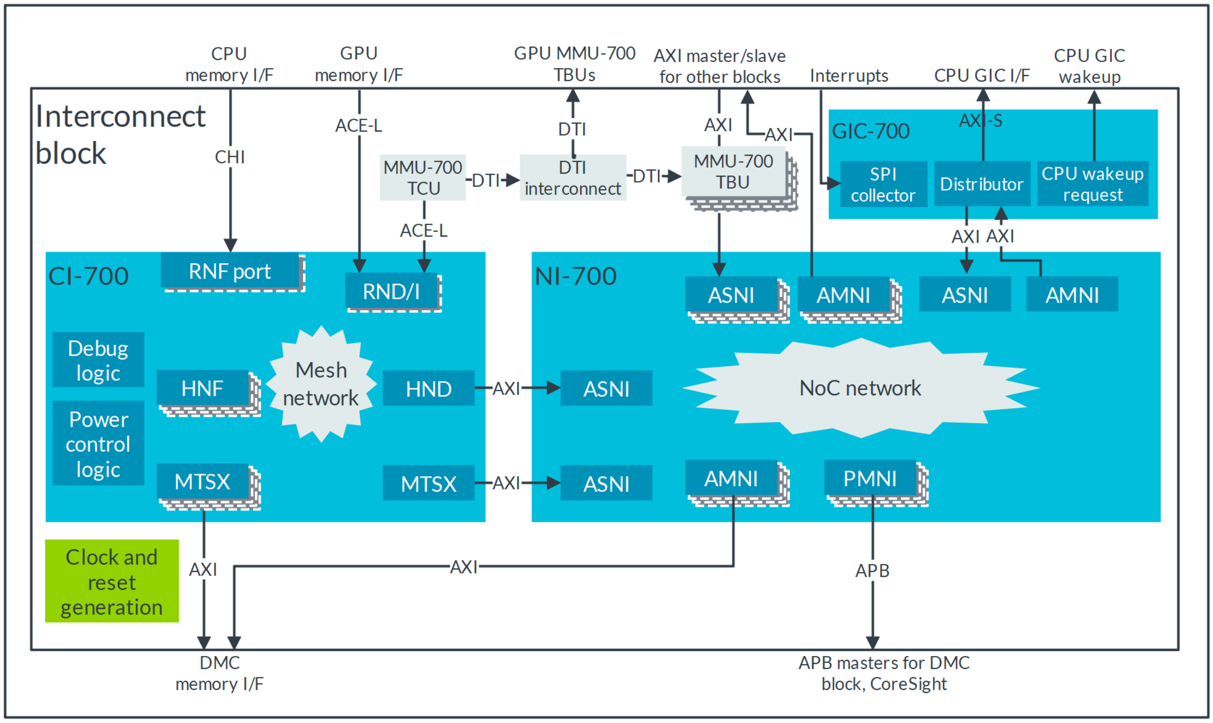
- Programmable system cache allocation hint for different type of GPU traffics through ACE AxCACHE to achieve optimal allocation policies for different workloads
- Easy integration
- Latency tolerance
- Compressed texture formats
- *Arm Fixed Rate Compression* (AFRC)
- *Arm Frame Buffer Compression* (AFBC) 1.3
- Configurable per-core power management for enabling the optimal power and performance combination for each application
- Coherency aware interconnects for system memory and resource sharing
- 8-bit, 10-bit, and 16-bit YUV input and output formats
- Secure processing of DRM-protected content

3.5 Interconnect block

The Interconnect block in the Total Compute 2021 Reference Design (RD-TC21) includes coherent and non-coherent interconnects, generic interrupt controller, and system memory management unit. The Interconnect block also has master and slave interfaces.

The following figure shows a high-level representation of the Interconnect block.

Figure 3-4: Block diagram of Interconnect block



3.5.1 Cache coherent mesh interconnect

The CoreLink™ CI-700 is a scalable configurable coherent interconnect that is designed to meet the *Power, Performance, and Area* (PPA) requirements of your SoC design.

For more information, see *Arm® CoreLink™ CI-700 Coherent Interconnect Technical Reference Manual*.

Features

The CI-700 provides the following features:

- Custom mesh size and device placement
- A programmable *System Address Map* (SAM)
- Optional *Component Aggregation Layer* (CAL) for device interface port expansion
- Optional support for non-XY routing algorithm between specified source-target pairs
- TxnID of 12 bits for all flits
- Maximum *Physical Address* (PA) width of 40 bits
- DVM message transport between masters
- QoS regulation for shaping traffic profiles
- Configurable QoS override to transactions targeting specific memory regions

- A *Performance Monitoring Unit* (PMU) to count performance-related events
- High-performance distributed *System Level Cache* (SLC) and *Snoop Filter* (SF):
    - The HN-F includes an integrated *Point-of-Serialization* (PoS) and *Point-of-Coherency* (PoC). The HN-F SLC also referred to as Agile System Cache that can be used both for compute and I/O caching.
- CHI *Memory Tagging Enhancements* (MTE)
- *Device Credited Slices* (DCSs) used for register slices at device interfaces, allowing flexibility in device placement
- *Mesh Credited Slices* (MCSs) used for X-Y register slices, allowing flexibility in mesh floorplanning
- *CAL Credited Slices* (CCSs), allowing flexibility in mesh floorplanning in configurations that use the CAL
- *On-Chip Memory* (OCM) allows for the creation of CI-700 systems without physical DDR memory
- RAS features including transport parity, optional data path parity, *Single-Error Correction and Double-Error Detection* (SECDED) ECC, and data poisoning signaling
- *Address Based Flush* (ABF)
- Way-based SLC partitioning
- Source-based way locking

3.5.2 Non-cache coherent interconnect

The CoreLink™ NI-700 is a highly configurable AMBA-compliant system level interconnect. NI-700 enables you to create a non-coherent interconnect that is optimized to the *Power, Performance, and Area* (PPA) requirements of your SoC design.

For more information, see *Arm® CoreLink™ NI-700 Network-on-Chip Interconnect Technical Reference Manual*.

Features

The NI-700 provides the following features:

- Native support for AMBA AXI protocol
- Native support for the following AMBA protocols:
    - AXI5, AXI-G, and AXI-H
    - AHB5
    - APB3 and APB4
    - AXI3, only on NI-700 master interfaces
- Packet transfer over multiple clock, power, and voltage domains
- Source-based packet routing
- Worm-hole routing with support for multiple *Resource Planes* (RPs)

- Flit-level credit-based flow control
- *Quality of Service* (QoS) features for prioritization of information transfer
- Distributed switching mechanism to enable traffic management and protect against network saturation
- Variable, user-defined topology that is specified through Socrates™ IP tooling platform

3.5.3 System Memory Management Unit

The CoreLink™ MMU-700 is a *System Memory Management Unit* (SMMU) that translates an input address to an output address.

For more information, see *Arm® CoreLink™ MMU-700 System Memory Management Unit Technical Reference Manual*.

Features

The MMU-700 provides the following features:

Compliance with the SMMUv3.2 architecture

- Support for stage 1 translation, stage 2 translation, and stage 1 followed by stage 2 translation

  Single stage, 52-bit

  Stage 1 translation that translates an input *Virtual Address* (VA) to an output *Physical Address* (PA)

  Two-stage, 52-bit

  The translations are:

  - Stage 1 translation that translates an input VA to an output *Intermediate Physical Address* (IPA)
  - Stage 2 translation that translates an input IPA to an output PA
- Support for Armv8 AArch32 and AArch64 translation table formats
- Support for 4KB, 16KB, and 64KB granule sizes in AArch64 format
- Support for MPAM
- Support for Secure EL2
- Masters can be stalled while a processor handles translation faults, enabling software support for on-demand paging
- Configuration tables in memory can support more than a million active translation contexts
- Queues in memory perform MMU-700 management. There is no requirement to stall a processor when it accesses the MMU-700.
- *A Performance Monitoring Unit* (PMU) in each *Translation Buffer Unit* (TBU) and *Translation Control Unit* (TCU) that enables MMU-700 performance to be investigated
- *Reliability, Serviceability, and Availability* (RAS) features for RAM corruption detection and correction

Support for AMBA® interfaces

- ACE5 Lite TBU transaction interfaces that support cache stash transactions, deallocating transactions, and cache maintenance
- An architected AXI5 extension that communicates per-transaction translation stream information
- An ACE5 Lite+*Distributed Virtual Memory* (DVM) TCU table walk interface that enables Arm®v8.5 processors to perform shared *Translation Look-aside Buffer* (TLB) invalidate operations without accessing the MMU-700 directly
- An ACE5 Low Power extension that enables the TCU to subscribe to DVM TLB invalidate requests on powerup and powerdown without reprogramming the *Direct Translation Interface* (DTI) interconnect
- AMBA DTI communication between the TCU and TBUs, enabling masters to request translations and implement TBU functionality internally
- Support for the AMBA *Low Power Interface* (LPI) Q Channel so that standard controllers can control power and clock gating
- AXI5 WAKEUP signaling on all interfaces, including DTI and APB interfaces
- Support for ACE5 Lite atomic transactions in the ACE Lite TBU
- Support for *Local Translation Interface* (LTI)
- Support for a dedicated *Generic Interrupt Controller* (GIC) integration, with *Message Signaled Interrupts* (MSIs) supported for common interrupt types

Support for flexible integration

- You can place a configurable number of TBUs close to the masters being translated
- Communication between TBU and TCU over AXI4 Stream is supported using the supplied DTI interconnect components, or any other AXI4 Stream interconnect
- DTI interconnect components support hierarchical topologies and control the tradeoff between the number of wires and the DTI bandwidth

Support for high performance translation

- Scalable configurable MicroTLB and *Main TLB* (MTLB) in the TBU can reduce the number of translation requests to the TCU
- TBU direct indexing and MTLB partitioning enable the use of MTLB entries to be managed outside the TBU, improving real time translation performance
- Optimization enables storage of all architecturally defined page and block sizes, including contiguous page and block entries, as a single entry in the TBU and TCU TLBs (WCs)
- Per TBU prioritization in the TCU enables high priority transaction streams to be translated before low priority streams
- TCU prefetch of translation tables, which can be enabled on a per context basis, improves translation performance for real time masters that access memory linearly
- *Hit Under Miss* (HUM) support in the TBU enables transactions with different AXI IDs to be propagated out of order, when a translation is available

- TBU detects multiple transactions that require the same translation so that only one TBU request to the TCU is required
- TCU detects multiple translations that require the same table in memory so that only one TCU memory request is required
- Multi-level, multi stage walk caches in the TCU reduce translation cost by performing only part of the table walk process on a miss
- A configurable number of concurrent translations in the TBU and TCU promotes high translation throughput

3.5.4 Generic Interrupt Controller

The CoreLink™ GIC-700 handles interrupts from peripherals to the cores and between cores. The GIC-700 provides interrupt services and masking, registers and programming, interrupt grouping, security, and performance monitoring.

For more information, see *Arm® CoreLink™ GIC-700 Generic Interrupt Controller Technical Reference Manual*.

Features

Interrupt services and masking

The GIC-700 provides the following interrupt features:

- Support for the following interrupt types:
    - Up to 1984 *Shared Peripheral Interrupts* (SPIs) in groups of 32.
    - Up to 48 *Private Peripheral Interrupts* (PPIs) that are independent for each core and can be programmed to support either edge-triggered or level-sensitive interrupts.
    - Up to 16 physical *Software Generated Interrupts* (SGIs) for each core, which the core generates through its GIC CPU interface.
- Interrupt masking and prioritization with 32 priority levels, 5 bits per interrupt.

Registers and programming

The GIC-700 provides the following programming features:

- Flexible affinity routing, using the *Multiprocessor Identification Register* (MPIDR) addresses, including support for four affinity levels (0-3).
- Single ACE5-Lite slave port on each chip for programming of all registers but excluding the GITS_TRANSLATER register in non-monolithic configurations. Each ITS has an optional ACE5-Lite slave port for programming the GITS_TRANSLATER register.
- Coherent view of SPI register data across multiple chips.

Security

The GIC-700 provides the following security feature:

- A global Disable Security signal. The gicd_ctlr_ds signal enables support for systems without security support.

For more information about Exception levels, see the *Arm® Architecture Reference Manual for A-profile architecture*.

Performance monitoring

The GIC-700 provides *Performance Monitoring Unit* (PMU) counters with snapshot functionality.

3.6 SCP block

The *System Control Processor* (SCP) block is an always-on block inside the design.

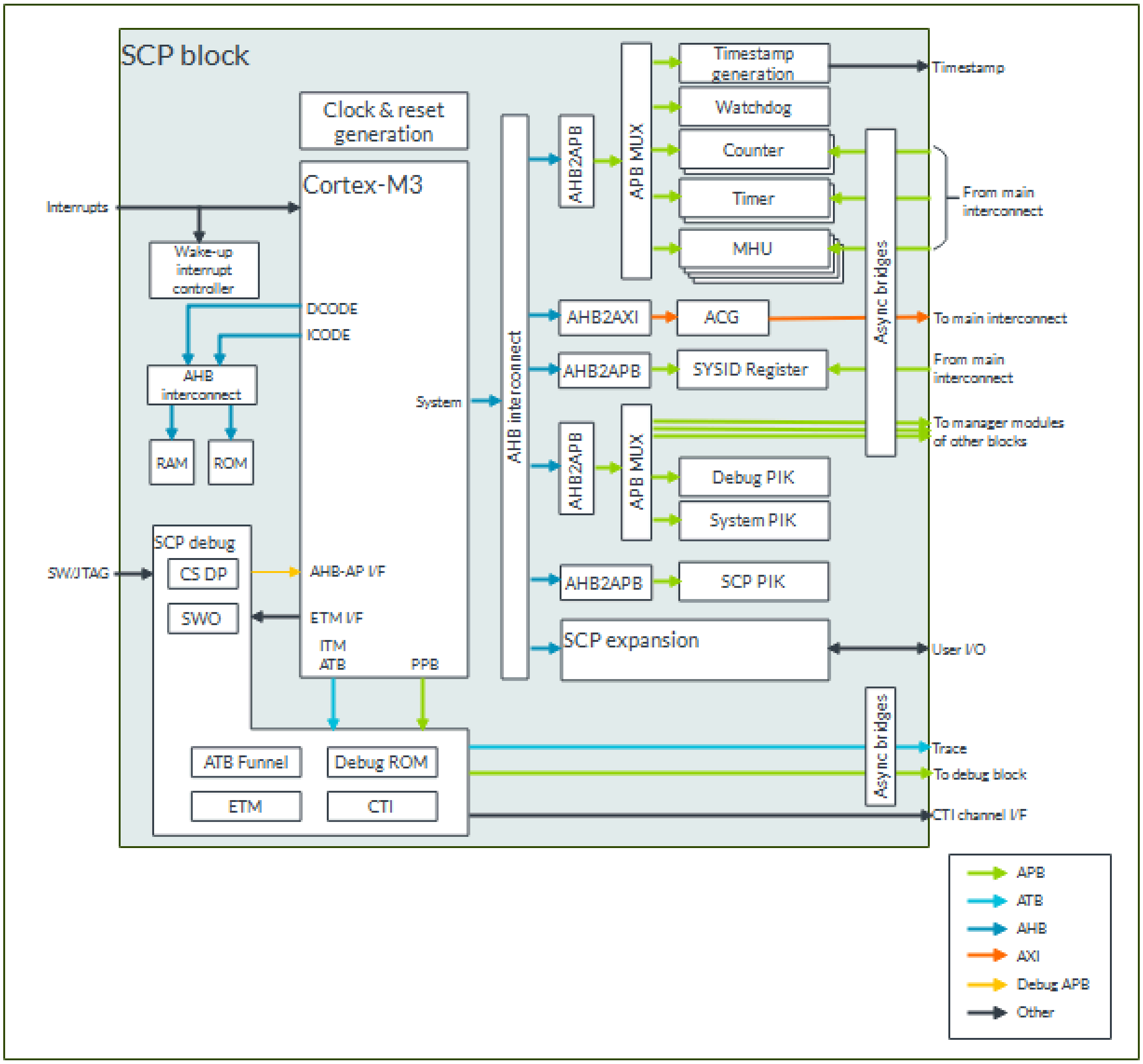
It is mainly responsible for low-level system management, including the following features:

- Boot and system start-up
- Root of the chain of trust for *Arm® Base System Architecture* compliance
- Initial static configuration
- Managing transitions between operating points, that is, 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

In addition, it 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 a high-level representation 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 peripherals (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. Each PIK is designed specifically for the power domains in the region it controls. For example, the Debug PIK controls the power and voltage domains inside the Debug block. Most PIKs incorporate *Power Control Units* (PPUs) from Arm® CoreLink™ PCK-600 for coordinating components in its targeted power and voltage domains.

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

Users who want to add SoC-specific storage interfaces and other SCP peripherals, such as flash controllers, I2C controllers, and SPI controllers, can use an expansion module. *Advanced High performance Bus* (AHB) data bus, interrupts, clock, reset, and power control signals (Q-Channels) are provided to the module.

Built with Arm® CoreSight-600 components, SCP debug connects to an external debugger through a *Serial Wire* (SW) or *Joint Test Action Group* (JTAG) interface. It provides full debug and trace capability to the Cortex-M3 processor. The SCP debug also acts as the debug entry point to the main Debug block at the system level. It connects to the Debug block through debug APB, *Cross Trigger Interface* (CTI) channel interface, and trace interface.

The SCP is considered intrinsically Secure inside the reference subsystem. It can access both Secure and Non-secure regions in the main (AP) memory space. As an AP is considered intrinsically Non-secure, it cannot directly access SCP memory space. Rather, it communicates with the SCP through a *Message Handling Unit* (MHU). This isolation provides security protection to the SCP from potential attacks from APs.

3.6.1 SCP components

The SCP peripherals region in the SCP memory map contains components such as Generic Timers, Generic Counters, Watchdog Timer, and Power registers.

Cortex®-M3 processor

The Cortex®-M3 processor is a high-performance, low-power 32-bit processor designed for the microcontroller market. It has the following key features:

- Outstanding processing performance combined with fast interrupt handling
- Enhanced system debug with extensive breakpoint and trace capabilities
- Efficient processor core, system, and memories
- Ultra-low power consumption with integrated sleep mode and an optional deep sleep mode
- Platform security robustness, with an optional integrated *Memory Protection Unit* (MPU).

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

- *Arm® Cortex®-M3 Processor Technical Reference Manual*

SRAM and Secure boot ROM

The SRAM and ROM are used by the Cortex®-M3 processor for basic operation.

SCP Watchdog

The SCP subsystem includes a Cortex-M System Design Kit Watchdog Timer that protects against lockups in the firmware.

For more information about the watchdog, see SCP Watchdog.

Counter and timer

Memory-mapped implementation of Arm generic counters and timers are running on REFCLK clocks. See Timers and counters for more information.

Message Handling Unit (MHU)

The MHU is a memory-mapped peripheral that provides a mechanism to assert interrupt signals to facilitate inter-processor message passing between the SCP and the AP. The message payload can be deposited into main memory or on-chip memories. Therefore, the MHU is simply used as a message signaling mechanism.

The MHU asserts the following interrupt signals:

- A high priority Non-secure interrupt and a low priority Non-secure interrupt in the AP interrupt map
- A Secure interrupt in the AP interrupt map
- A high priority Non-secure interrupt and a low priority Non-secure interrupt in the SCP interrupt map
- A Secure interrupt in the SCP interrupt map

See Interrupt maps for RD-TC21 for the interrupt maps of the AP and SCP.

For each of the interrupt signals, the MHU drives the interrupt using a set of registers to enable software to set, clear, and check the status of each. For the MHU programmers model, see Message Handling Unit registers.

Power control registers

These registers manage the power domain state transitions that the SCP firmware controls. See Power control registers.

3.6.2 Sleep modes

The Processor SLEEP modes reduce power consumption.

The *System Control Processor* (SCP) has the following sleep modes:

SLEEPING

In the SLEEPING mode, the SCP gates the clock to the Cortex®-M3 processor, but other logic in the SCP continues to run. The REFCLK remains available. The SCP will exit SLEEPING when an interrupt or event is received by the Cortex®-M3 processor.

SLEEPDEEP

In the SLEEPDEEP mode, the SCP can de-assert its request for REFCLK, that is, REFCLKQACTIVE goes low, when the following conditions are met:

- The REFCLKFORCE bit is not set in the SCP Power Control registers
- The SCP *Debug Access Port* (DAP) is not connected to the debugger

Software must save the REFCLK domain before entering into the SLEEPDEEP mode when the WICEN_STATUS field is 0. The SCP will only exit SLEEPDEEP when an interrupt is received on one of the SCP wakeup interrupts.

The SLEEPDEEP bit in the Cortex-M3 processor system control register determines the sleep mode that the Cortex-M3 processor enters:

- When the SLEEPDEEP bit is set and the processor enters sleep, the Cortex-M3 processor enters the SLEEPDEEP mode.
- When the SLEEPDEEP bit is clear and the processor enters sleep, the Cortex-M3 processor enters the SLEEPING mode.

The SCP firmware should only enable entry to the SLEEPDEEP mode when the rest of the SoC has been powered down.

The following table shows the mapping between:

- The SCP sleep modes
- The value of the WICEN_STATUS field of the WIC based Deep Sleep Status register in the SCP Power Control registers.
- The legal subsystem power modes

Table 3-3: Mapping between SCP power modes and subsystem power modes

Cortex-M3 sleep mode	WICEN_STATUS	Power Modes
SLEEPING	X	CSS.RUN/CSS.SLEEP0
SLEEPDEEP	1	CSS.SLEEP0
SLEEPDEEP	0	CSS.SLEEP1

3.7 Memory block

The Memory block contains a memory controller that implements data path from the cache-coherent interconnect. The Memory block wraps the memory controller with the necessary glue-logic to integrate with the rest of the subsystem.

It contains the inline external TrustZone controller TZC-400, an Address translation function, power management, clock and reset generation, and control logic.

3.7.1 TZC-400 Address Space Controller

The Memory block includes CoreLink™ TZC-400 TrustZone® Address Space Controller in the memory data path between interconnect ports and *Dynamic Memory Controller* (DMC) controller.

TZC-400 performs security checks on transactions to memory. You can use the TZC-400 to create up to eight separate regions in the address space, each with an individual security level setting. Any transactions must meet the security requirements to gain access to the memory. You can program the base address, top address, enable, and security parameters for each region.

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

Features

TZC-400 provides the following features:

- The ability to define up to eight address regions in the address map
- A default base region to cover all remaining portions of the address map
- Software programmable security access permissions for each address region through an *Advanced Peripheral Bus 4* (APB4) interface. This includes the default base region, Region 0.
- Configure to one Filter unit to only allow data transfer between an *AXI Coherency Extensions Lite* (ACE-Lite) master and an ACE-Lite slave if the security status of the ACE-Lite transaction and its identity match the security settings of the memory region it addresses.
- Common region configuration register settings that are shared between multiple filter units
- The filter units can support asynchronous clocks that are independent of each other and of the control unit that is clocked by the *Advanced Peripheral Bus* (APB) interface clock.
- Dual read access path
    - Fast path for low-latency accesses but with limited outstanding access support
    - Normal path for accesses with a much higher outstanding access support
- Identity-based filtering of Non-secure accesses
- Reporting and interrupt signaling that is configurable from software to manage failed permission checks
- Speculative accesses to support *Quality of Service* (QoS) *Virtual Network* (QVN) and fast path. This feature can be disabled
- *Advanced eXtensible Interface* (AXI) low-power interface for each clock domain
- Gate keeper to allow or block accesses to each filter unit
- Support for 256 outstanding transactions on the normal paths

There are four instances of TZC-400 each with:

- Two Filter Units, one for CoreLink™ CI-700 and one for CoreLink™ NI-700 interface of the Dynamic Memory Controller. These filter units share the same region configurations.
- Nine regions, that include one base region, region 0, and eight fully software configurable regions, that is, region 1-8. These enable up to eight independent regions with different security requirements to be defined.

3.7.1.1 Address translation logic

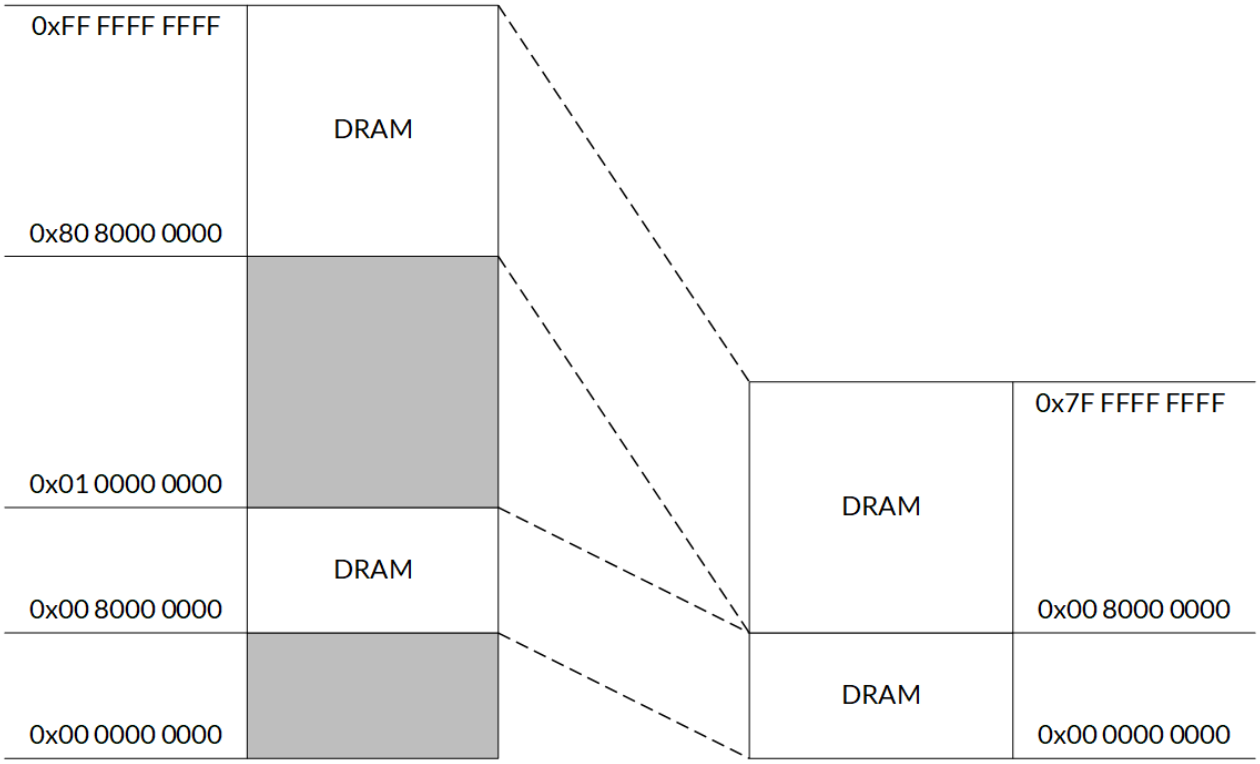
The address translation converts the system address space to a contiguous address space for each DMC.

The address translation is performed in three stages:

Stage 1

Non-DRAM regions are removed from the address as shown in the following figure:

Figure 3-6: Memory block address translation



Stage 2

Shuttering accounts for the effect of hashing between multiple DMCs.

Stage 3

The bank bits in the resulting address are hashed with the row data.

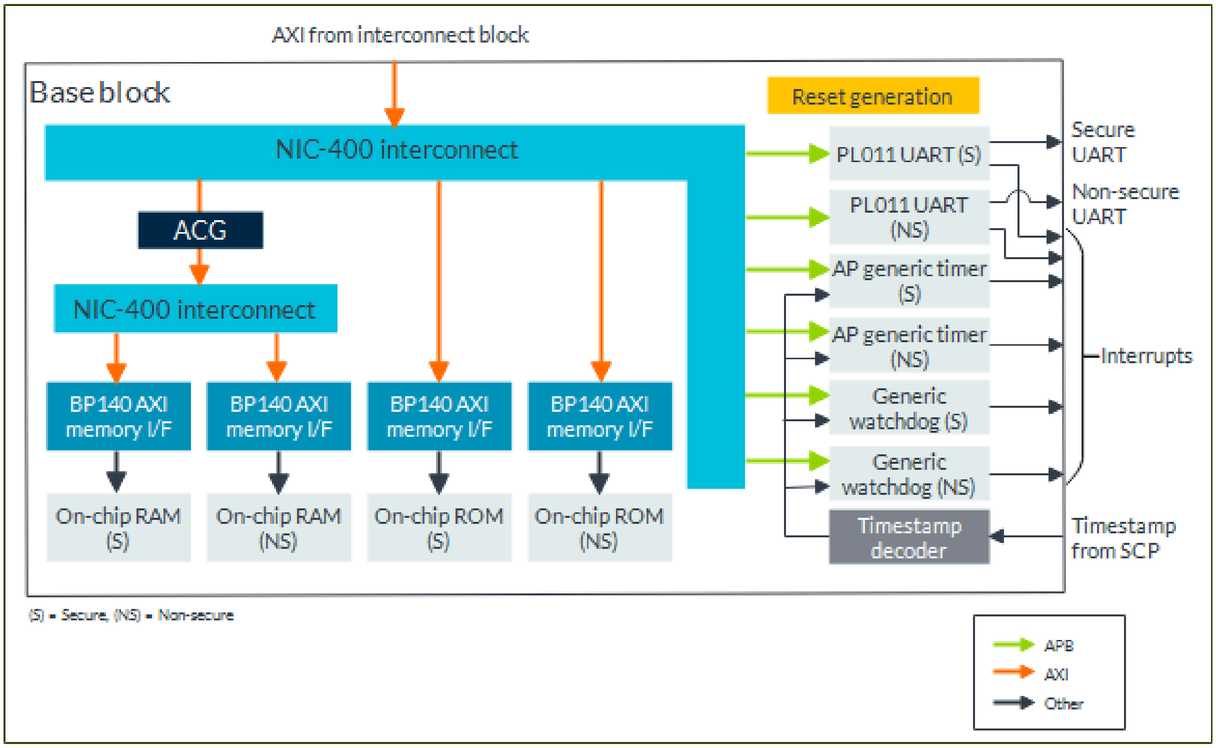
All other bits of output are directly connected to the input.

3.8 Base block

The Base block contains on-chip memory (RAM and ROM) and system peripherals for *Application Processors* (APs). The Base block also includes a CoreLink™ NIC-450 interconnect, which the peripherals connect to.

The following figure shows a high-level representation of the Base block.

Figure 3-7: Block diagram of Base block



The CoreLink™ NIC-450 includes all CoreLink™ NIC-400 features.

As shown, APs can access memory and peripherals through the *Advanced eXtensible Interface* (AXI) connected to the main Interconnect block. 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) in the diagram is for supporting 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.

A full list of memories and peripherals contained in the block is given in the following table.

Table 3-4: Memories and system peripherals of base block

Name	Description
Secure ROM	This 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	This 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 generic watchdog	Generic watchdog timer for Secure applications
Non-secure generic watchdog	Generic watchdog timer for Non-secure applications
Secure AP generic timer	Generic for Secure applications
Non-secure AP generic timer	Generic for Non-secure applications

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

The CoreLink™ NIC-450 includes all CoreLink™ NIC-400 features. For more information, see the following documentation:

- *Arm® CoreLink™ NIC-450 Network Interconnect Technical Overview*
- *Arm® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual*

4. Functional description

Components in Total Compute 2021 Reference Design (RD-TC21), such as clocks and resets, work across multiple functional blocks.

4.1 Clock

Clocks in RD-TC21 are categorized according to their types and the functional blocks they are used in.

4.1.1 Input clocks

Input clocks must be provided with a source that is external to, or embedded in, the subsystem.

Table 4-1: Input clocks

Clock	Description
REFCLK	Main reference clock The input clock to *System Control Processor* (SCP) boots coming out of reset. REFCLK also clocks other logic like SCP generic and watchdog timers.
CPUxPLLCLKn	CPUxPLLCLKn is a high-frequency reference clock inputs used for application core PLL, that is, one per core, where n = 0-7.
CLUS0PLLCLKn	CLUS0PLLCLKn is a high-frequency reference clock inputs used for cluster PLL, that is, one clock per cluster, where n is the number of clusters in a subsystem. As RD-TC21 reference subsystem only contain one cluster, n is fixed to 1.
INTPLLCLK	INTPLLCLK is a high-frequency reference clock input used for the Interconnect block.
SYSPLLCLK	SYSPLLCLK is a high-frequency clock input for the main block.
DMCPLLCLK	DMCPLLCLK is a high-frequency reference clock input used for the Memory block.
GPUPLLCLK	GPUPLLCLK is a high-frequency reference clock input used for the GPU block.

4.1.2 Systems clocks

RD-TC21 internally derives clocks that are used for parts of the subsystem. These clocks are generated only when the VSYS.SYSTOP power domain is powered. Each clock can be individually controlled.

The following table summarizes these derived internal clocks.

Table 4-2: Systems internal derived clocks

Clock Signal	PLL	Description
CLUSxCORECLKn n is the number of clusters in a subsystem.	CPUPLL	Core clock. Each core is clocked independently to each other. Therefore, One clock per cluster, where
CLUSxCLK	CLUSxPLLCLKn	CPU Cluster x clock. The cluster clock drives the logic in the Processor block, where x is 0 to 31.
INTCLKOUT	INTPLLCLK	Coherent Interconnect Clock Clock for the interconnect and other expansion master and slave ports. This clock is generated from INTPLLCLK
DMC_1x_CLKn	DMCPLLCLK	Dynamic Memory Controller Clock Clock for the Dynamic Memory Controllers in the Memory block. Where, n is 0 - 7 for 8 DMC channels.This is generated from DMCPLLCLK.
DMC_2x_CLKn	DMCPLLCLK	Dynamic Memory Controller Clock This clock runs at twice the frequency of DMC1xCLKn. Where, n is 0 - 7 for 8 DMC channels. This is generated from DMCPLLCLK. If the external PHY does not have internal PLL to generate this clock, then you can connect DMC2xCLKn to the PHY.
DFICLKn	DMCPLLCLK	DFI interface clock Clocks the DFI interface between DMC and external PHY. This is generated from DMCPLLCLK.
SYSPERCLK	SYSPLLCLK	Interconnect Clock Clocks the NIC-400 switch in the Base block. This is generated of SYSPLLCLK.
SYSPCLKDBG	SYSPLLCLK	Debug APB Clock Clocks the APB interfaces and bridge at the top level
ATCLKDBG	SYSPLLCLK	Debug ATB Clock Clocks the ATB interfaces from the Debug block
GICCLK	SYSPLLCLK	GIC Clock Clocks the GIC-700 Generic Interrupt Controller
SCPCORECLK	SYSPLLCLK	SCPCOREClock Clocks the core of the SCP (this clock is sourced by REFCLK initially during full Subsystem reset, later switched over to divided down version of SYSPLLCLK by SCP firmware once the PLL is locked)
PCLKSCP	SYSPLLCLK	SCP APB Clock Clocks the APB interfaces from the SCP element

These clocks are asynchronous to each other and can be managed independently using programmer visible bits in the control registers in the reference subsystem:

Note

The cluster clock can be synchronous to the interconnect clock or core clock depending on whether it is synchronous to the interconnect or core.

- Clock source selection
- Clock division with all integer ratios between 1 and 16 supported
- Clock force enabled
    - When a clock is force-enabled, it is always switched on. This mode is intended for use when debugging clock control software. Some clocks do not have this mode, because they do not have any transparent hardware clock gating. In these cases, this mode is not required.

RD-TC21 provides glitch-less clock switching and selection, and clock dividers produce a clock 50:50 duty cycle.

4.1.3 Output clocks

Output clocks are source clocks for peripherals and other components that are integrated into *System on Chips* (SoCs) external to the system.

Table 4-3: Output clocks

Clock	Description
ACE-Lite slave, AXI master, and DMC DFI interface clocks	The AXI master, ACE-Lite slave expansion, and DMC DFI interfaces each have a clock output associated with them.
GICCLKOUT	The *Generic Interrupt Controller* (GIC) clock is exported externally.
GICCLKOUT_FREE	Free running GIC clock
SYSPERCLKOUT	System peripheral clock output. The output clock for expansion AXI4 master interface, EAXIMn.
SYSERCLKOUT_FREE	System peripheral clock output Free running output clock for expansion AXI4 master interface, EAXIMn.
SCP AXI expansion	An AXI interface for connection of the *System Control Processor* (SCP) peripherals. This interface is synchronous to the SCPAXICLK clock domain.

4.2 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.

4.2.1 REFCLK time domain

The *Application Processors* (AP) 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 *System Control Processor* (SCP).

A custom counter component, REFCLK Counter, generates a time value for the REFCLK time domain. The component meets the requirements of the memory-mapped counter module that the *Arm® Architecture Reference Manual for A-profile architecture* describes. It is in the VSYS.AONTOP power domain.

This REFCLK 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 the *Arm® Architecture Reference Manual for A-profile architecture* defines. The interrupts from these timers are mapped to *Private Peripheral Interrupts* (PPI) through the Generic Interrupt Controller. 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 more REFCLK generic timers, one Secure and one Non-secure, for use by the applications processors.
- 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, making the REFCLK time domain unavailable.

After restoring REFCLK, and before turning a component that can observe the REFCLK time domain into the ON power mode, the SCP firmware must ensure a consistent time value is set in the REFCLK Counter.

4.2.2 REFCLK counter

The REFCLK counter is an implementation of the memory-mapped counter module.

The *Arm® Architecture Reference Manual for A-profile architecture* defines the memory-mapped counter module. The counter is visible in both the *Application Processor* (AP) and *System Control Processor* (SCP) memory maps, labeled REFCLK CNTControl and REFCLK CNTRead in the AP memory map.

The counter is implemented in the VSYS.AONTOP power domain.

The following table shows a single frequency mode of the REFCLK counter.

Table 4-4: REFCLK counter frequency mode

Mode	Name
0	REFCLK frequency mode

The frequency mode table entry for the REFCLK frequency mode is writeable. It is intended that firmware initializes this entry during the boot sequence.

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

4.2.3 Timers

There are different timers used in Total Compute 2021 Reference Design (RD-TC21) reference subsystems. This subsection describes them grouped into two types, generic timers and watchdog timers.

4.2.3.1 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 *Arm® Architecture Reference Manual for A-profile architecture* defines the timer that conforms to the Arm generic timer.

The timer implements the following features:

- REFCLK CNTCTL
- REFCLK CNTBase0
- A single timer frame without a second view and no virtual timer capability

4.2.3.2 AP REFCLK generic timer

The Base block includes two memory-mapped Arm Generic Timers, which are defined by the *Arm® Architecture Reference Manual for A-profile architecture*, for general-purpose functions. 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.

4.2.3.3 Watchdog timers

This section describes the RD-TC21 watchdog timers.

4.2.3.3.1 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 about the programmers model of the Generic Watchdog, see Application Processor memory map.

4.2.3.3.2 SCP Watchdog

The *System Control Processor* (SCP) subsystem includes a Cortex®-M System Design Kit Watchdog Timer that protects against lockups in the firmware.

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.

4.2.3.3.3 Trusted Watchdog

The *Application Processor* (AP) subsystem 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 global reset does not preserve the Trusted Watchdog state through reset because this is not a requirement of *Client Base System Architecture*.

This watchdog increments by one every REFCLK cycle.

4.2.3.3.4 Watchdog security

The *System Control Processor* (SCP) firmware and Secure Watchdog Timers are accessible by Secure accesses only.

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

4.2.4 Processor core power down modes

A core can generate timer interrupts after entering into *Wait For Interrupt* (WFI) or *Wait for Event* (WFE) mode. Cores can enter WFI and WFE modes without any side effects relating to their timers.

When a cluster is powered down, the Generic Timer state is lost, so extra steps must be taken.

There are two models for powering down a processor core, referred to as AP Wakeup and SCP Wakeup.

4.2.4.1 AP Wakeup model

In the *Application Processor* (AP) Wakeup model, software running on the APs must ensure that no timers are active on the cluster that is to be powered down. The cluster can then be powered down by making a request to the *System Control Processor* (SCP).

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

4.2.4.2 SCP Wakeup model

This model applies when powering down the last *Application Processor* (AP).

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

You can then powerdown the processor by making a request to the *System Control Processor* (SCP). The AP_REFCLK Generic Timers cause the processor to be woken 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 *Generic Interrupt Controller* (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 restores state to the GIC. The interrupt causes a wakeup interrupt for the processor, and the SCP handles interrupt.

4.2.4.3 Memory map frames

The *Application Processor* (AP) and *System Control Processor* (SCP) memory map contains several frames that relate to the counter and timer components.

These frames are as follows:

- REFCLK CNTControl is the frame that contains the control registers for the REFCLK counter.
- REFCLK CNTRead is the frame that contains the status registers for the REFCLK counter.
- REFCLK CNTCTL is the CNTCTLBase frame for the SCP_REFCLK timer.
- REFCLK CNTBase0 is the CNTBase0 frame for the SCP_REFCLK timer.
- CS CNTControl is the frame that contains the control registers for the CoreSight timestamp counter.
- AP_REFCLK CNTCTL is the CNTCTLBase frame for the AP_REFCLK timer.
- AP_REFCLK_S CNTBase1 is the CNTBase1 frame for the AP_REFCLK timer, and is only accessible using Secure accesses.
- AP_REFCLK_NS CNTBase0 is the Non-secure CNTBase0 frame for the AP_REFCLK timer, and is accessible using both Secure and Non-secure accesses.

4.3 Power control

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

An RD-TC21 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 modes 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

4.3.1 Voltage domain

A voltage domain is defined as a collection of design elements supplied by a single voltage. The voltage supply to the domain might be scaled or switched off for power or performance reasons.

Total Compute 2021 Reference Design (RD-TC21) supports 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 GPU. Supports DVFS.

VSYS

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

4.3.2 Power domain

A power domain is a collection of design elements within a voltage domain that share a common power strategy. A voltage domain can have 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.

The following table summarizes the supported top-level power domains within the VCPUn, VSYS, and VGPU voltage domains.

Table 4-5: Top-level power domains

Name	Implemented as	Voltage Domain	Power-gated?	Description
DBGTOP	Power-gated domain	VSYS	Yes	The power domain for the Debug related logic
AONTOP	Always-active region	VSYS	No	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	Power-gated domain	VCPU1	Yes	The power domain for the Processor cluster
GPUTOP	Power-gated domain	VGPU	Yes	The power domain for GPU
SYSTOP	Power-gated domain	VSYS	Yes	The power domain for the rest of the subsystem

Where:

- DBGTOP is the power domain for the 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 GPU. This is a power-gated domain.
- SYSTOP is the power domain for the rest of the system. This is a power-gated domain.

Within each top-level power domain, there are additional lower-level power domains defined by the respective blocks. There are corresponding CSYSPWRUPREQ signals for each of the power domains that enable debug logic to request powering up the respective power domains. All the signals are driven from the CoreSight design information.

GIC power domain

The *Generic Interrupt Controller Distributor* (GICD) will be in the SYSTOP power domain.

The *Interrupt Translation Service* (ITS) blocks will be in the same power domain as the respective PCIe master power domain.

The *Peripheral Port Interfaces* (PPIs) will be in the respective cluster power domain.

4.3.3 Domain hierarchy

A *Power Policy Unit* (PPU) controls each power domain. PPUs contain a programmer-visible state that the SCP uses to set the power policy and control the power mode of the power domains.

A hierarchy of power domains exists in the subsystem.

The voltage and power domain hierarchies in this section are illustrative of the logical relationships between the power-gated domains.

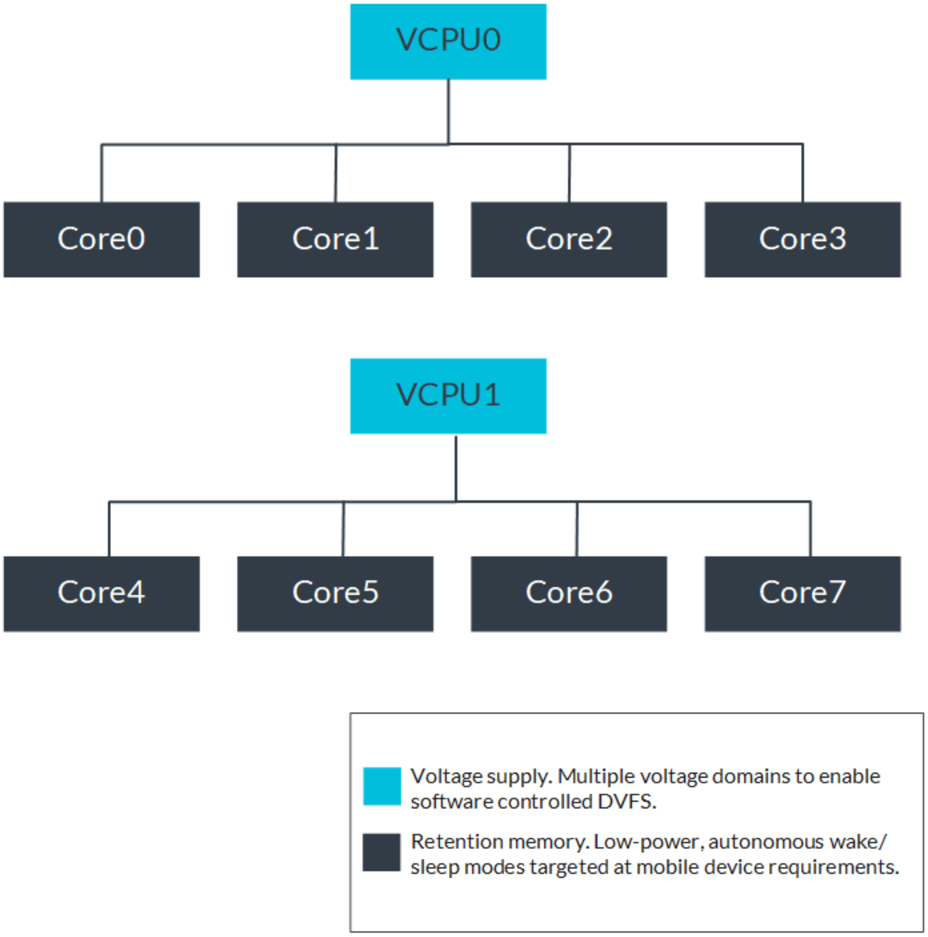
For more information about the PPU, see the *Arm® Power Policy Unit Architecture Specification, version 1.1*.

4.3.3.1 VCPUn

The Processor block lies in the VCPUn voltage domain. Where n is 0-1.

The following figure shows the logical relationships between the power-gated domains within the VCPUn voltage domain for all configurations.

Figure 4-1: VCPUn domain hierarchy

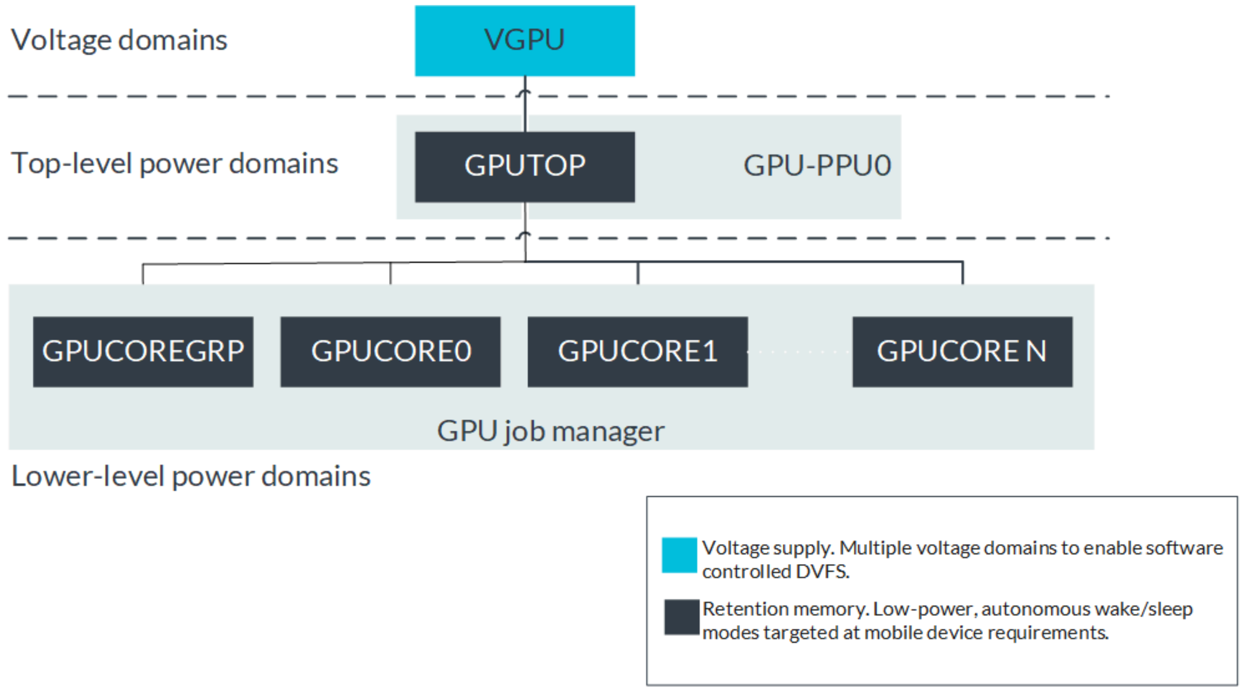

VCPU0
Core0
Core1
Core2
Core3
VCPU1
Core4
Core5
Core6
Core7
Voltage supply. Multiple voltage domains to enable software controlled DVFS.
Retention memory. Low-power, autonomous wake/sleep modes targeted at mobile device requirements.

4.3.3.2 VGPU

The GPU block lies in the VGPU voltage domain.

The following figure shows the superset of supported power-gated domains, always active regions, and retention memories within the VGPU voltage domain. The hierarchy illustrates only the logical relationships between the power-gated domains.

Figure 4-2: VGPU domain hierarchy


Voltage domains
VGPU
Top-level power domains
GPUTOP
GPU-PPU0
GPUCOREGRP
GPUCORE0
GPUCORE1
GPUCORE N
GPU job manager
Lower-level power domains
Voltage supply. Multiple voltage domains to enable software controlled DVFS.
Retention memory. Low-power, autonomous wake/sleep modes targeted at mobile device requirements.

The GPUTOP power domain contains the job manager and all necessary integration logic to incorporate the GPU into a dedicated voltage domain.

The GPUCOREGRP region incorporates a core group consisting of L2 cache and infrastructure logic in support of a cluster of shader cores.

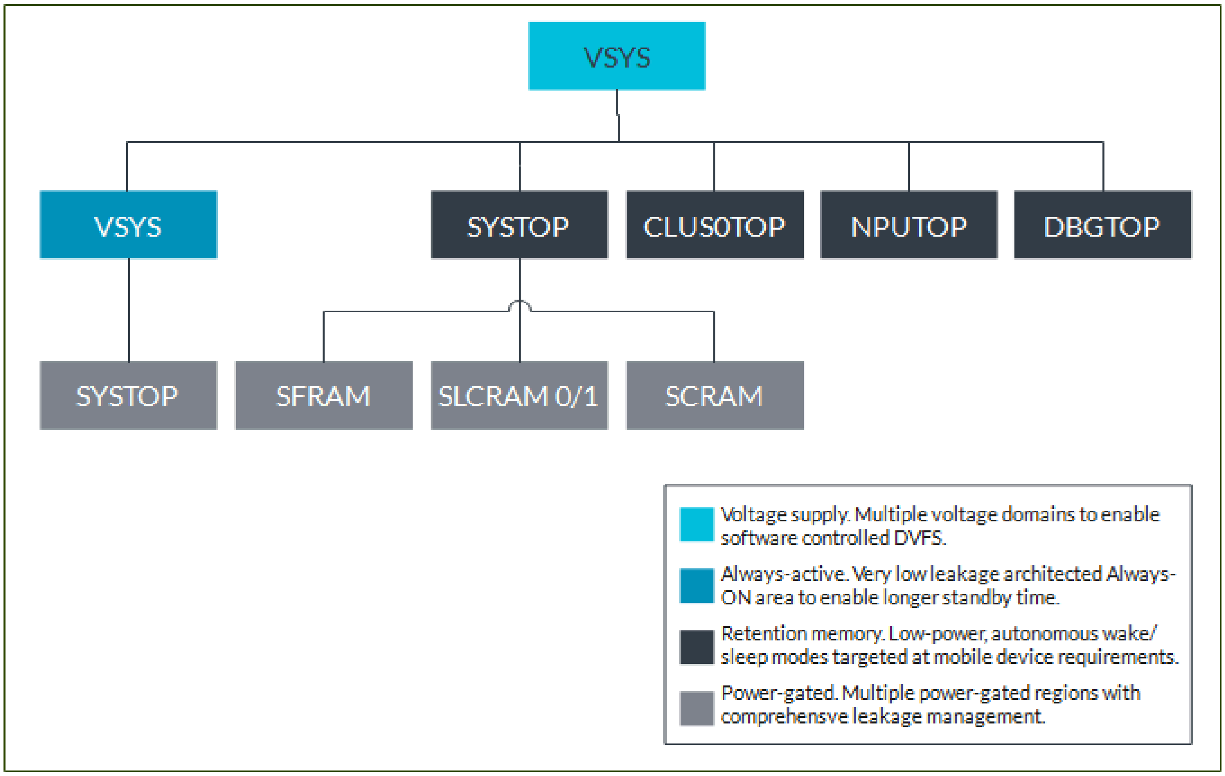
The GPUTOP region is managed by the SCP. All other power domains in the VGPU are managed by GPU driver which is responsible for low-level power control and scheduling.

4.3.3.3 VSYS

The rest of the subsystem lies in the VSYS voltage domain.

The following figure shows the logical relationships between the power gated domains within the VSYS voltage domain.

Figure 4-3: VSYS domain hierarchy


VSYS
VSYS
SYSTOP
CLUS0TOP
NPUTOP
DBGTOP
SYSTOP
SFRAM
SLCRAM 0/1
SCRAM
Voltage supply. Multiple voltage domains to enable software controlled DVFS.
Always-active. Very low leakage architected Always-ON area to enable longer standby time.
Retention memory. Low-power, autonomous wake/ sleep modes targeted at mobile device requirements.
Power-gated. Multiple power-gated regions with comprehensve leakage management.

4.4 Security

Total Compute 2021 Reference Design (RD-TC21) supports Level 1 of the *Trusted Base System Architecture, Client (4TH Edition)*.

4.4.1 Interconnect block security

The Interconnect block supports the transfer of the Non-secure flag from all master interfaces to slave interfaces either through the Request Flit on the *Coherent Hub Interface* (CHI), or through AxPROT[1] bit on ACE-Lite, AXI4, and AXI3 interfaces. These define the security settings of an access.

The REQFLIT.NS or AxPROT[1] field specifies whether a read/write transaction is Secure or Non-secure.

Within RD-TC21, peripheral accesses are placed into Secure or Non-secure world either by:

- The peripheral itself, interpreting the Non-secure flag.
- Permanently as Secure or Non-secure, or software-configurable using the programmers model registers of the interconnect.

4.4.2 Base block security

The Arm® CoreLink™ NIC-400 Network Interconnect within the Base block supports the transfer of the *Non-secure* (NS) flag from all master interfaces to slave interfaces through the AxPROT[1] bit on ACE-Lite, AXI4, and AXI3 interfaces.

The AxPROT[1] bit on these interfaces defines the security setting of an access.

The REQFLIT.NS or AxPROT[1] field specifies whether a read/write transaction is Secure or Non-secure.

Within RD-TC21, peripheral accesses are placed into Secure or Non-secure world either by:

- The peripheral itself, interpreting the Non-secure flag.
- NIC-400, either permanently as Secure or Non-secure, or software-configurable using the programmers model registers of the CoreLink™ NIC-400.

4.4.3 SCP block security

It is essential that malware does not compromise the block, because the *System Control Processor* (SCP) block has access to clock, resets, power, and peripheral register states.

Therefore, the block, including its internal RAM and ROM memories, is treated as inherently trusted.

- The SCP boots from its private on-chip trusted SCP ROM so that the boot code cannot be modified.
- The main SCP firmware is initially placed in an external device, for example, flash, and the application processor must authenticate it before it is copied to the SCP on-chip trusted RAM. To facilitate this, the application processor must be able to start up before the main firmware for the SCP is available.
- Any access from the SCP block to the rest of the system is trusted.

Any Secure data that the SCP stores, for example, data related to Secure state save and restore operations, must be stored in one of the following ways:

- Its own internal private SRAM that is not accessible from outside the block.
- In areas that are Secure access only, for example, the trusted on-chip SRAM.

4.5 System boot

Total Compute 2021 Reference Design (RD-TC21) reference subsystems support Secure boot.

To support Secure boot, the subsystem provides:

- A Cortex®-M3-based *System Control Processor* (SCP) is designed to function as a Trusted block. This subsystem includes:
    - A local on-chip Trusted boot ROM
    - A host that loads the SCP runtime firmware from the external non-volatile memory to the internal secure RAM. The host signals the SCP to copy the runtime firmware into the SCP RAM. The copying is performed by the SCP ROM code.
    - A local on-chip Secure SRAM to execute the main SCP firmware loaded in from external non-volatile memory
- A Cortex®-A series-based system with the following features:
    - On-chip Secure boot ROM intended for storing Trusted boot code
    - On-chip Non-secure ROM intended for storing Non-secure boot code. This feature is mainly used for the user-defined implementation settings.
    - On-chip Secure SRAM intended for storing Trusted data

5. FVP

A *Fixed Virtual Platform* (FVP) enables the development of software without the requirement for the prototype hardware. Arm FVP models use binary translation technology to deliver fast simulations of the Arm-based system.

The FVP drives system architecture and software standardization. It enables efficient software and firmware development reducing the amount of work that is required to bring up a complete system.

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

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

Reference Configuration

The Total Compute 2021 Reference Design (RD-TC21) FVP models many of the Arm® IP involved in the subsystem, and supports the following configuration:

T1 Mobile I

1xCortex-X3 3xCortex-A715 4xCortex-A510, Mali-G715 GPU, Memory Controller 4x16-bit Memory interfaces

The FVP is used with the software package. See Arm® Total Compute solution for instructions on Software, how to set up and run the FVP.

5.1 About Total Compute 2021 Reference Design (RD-TC21) FVP

The Total Compute 2021 Reference Design (RD-TC21) Fixed Virtual Platform (FVP) models T1 Mobile I reference subsystems.

The FVP models the following IP components inside the subsystem:

- Arm® Cortex®-A715 Core
- Arm® Cortex®-X3 Core
- Arm® Cortex®-A510 Core
- Arm® DynamIQ™ Shared Unit-110
- Arm® CoreLink™ CI-700 Coherent Interconnect
- Arm® CoreLink™ NI-700 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™ PCK-600 Power Control Kit
- On-Chip ROM, RAM, and other peripherals
- Clock generators

The FVP does not model the following RD-TC21 subsystem IP components:

- Arm® Mali™-G715 GPU
- Arm® CoreSight™ System-on-Chip SoC-600

The FVP is used with the software package. See Arm® Total Compute solution for instructions on Software, how to set up and run the FVP.

5.2 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 2021 Reference Design (RD-TC21) reference subsystem. These components create the necessary environment for running and developing a reference software stack.

The Total Compute 2021 Reference Design (RD-TC21) *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 which contains the SoC

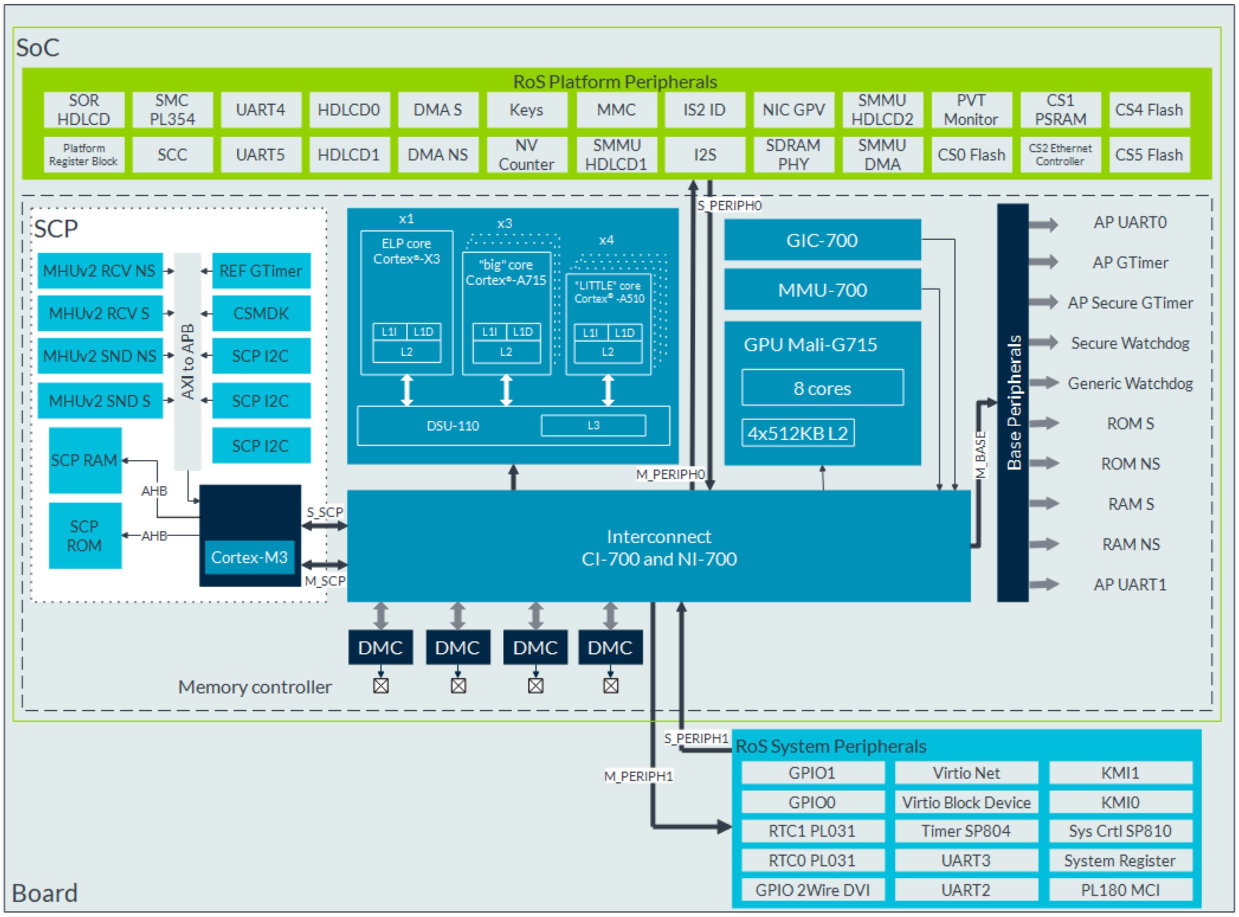
The last two of these three parts are covered by the RoS. Similar to how the CSS should be integrated in an SoC and board, the RoS components connect to the CSS through its expansion interfaces. This adds to the expansion sections of the memory map and interrupt map of the CSS.

For more information, see the following sections:

- Memory maps for RD-TC21
- Interrupt maps for RD-TC21

The following figure shows the RoS and its peripherals connection.

Figure 5-1: RoS and peripheral connections



Remember

The FVP does not model the following RD-TC21 subsystem IP components:

- Arm® Mali™-G715 GPU
- Arm® CoreSight™ System-on-Chip SoC-600

The following table summarizes the peripheral IP.

Table 5-1: Peripheral 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
TRNG	Registers for True Random Number Generator
NVCounter	Registers for non-volatile counter
Keys	Register block for Secure keys

IP name	Function
UART	Test output
PL180 MCI	Multimedia card interface
RTC	Real time clock
SP805	Watchdog
SP804	Dual timer
KMI	Keyboard and mouse interface
GPIO	General purpose I/O, PL061_GPIO
SOR HDLCD	System Override registers
Platform register block	Platform ID register
I2S	I2S implemented by PL061_GPIO
PSRAM	Pseudostatic DRAM
Ethernet Controller	SMSC 91C111
VirtioBlockDevice	Virtio block device model

5.3 FVP peripherals

The RD-TC21 FVP includes peripherals that the software payload requires to run.

These peripherals are organized in layers as follows.

5.3.1 Board peripherals memory map

The following table summarizes the RoS memory map for system peripherals in terms of offsets from `0x000_1C00_0000` within a 2MB memory range.

Table 5-2: RoS system peripherals memory map

Start Address Offset	End Address Offset	Name	Size
`0x1F_0000`	`0x1F_FFFF`	Reserved	64KB
`0x1E_0000`	`0x1E_FFFF`	GPIO1	64KB
`0x1D_0000`	`0x1D_FFFF`	GPIO0	64KB
`0x19_0000`	`0x1E_FFFF`	Reserved	256KB
`0x18_0000`	`0x18_FFFF`	RTC1 PL031	64KB
`0x17_0000`	`0x17_FFFF`	RTC0 PL031	64KB
`0x16_0000`	`0x16_FFFF`	GPIO 2 Wire(DVI)	64KB
`0x15_0000`	`0x15_FFFF`	Virtio Net	64KB
`0x14_0000`	`0x14_FFFF`	Reserved	64KB
`0x13_0000`	`0x13_FFFF`	Virtio Block Device	64KB
`0x12_0000`	`0x12_FFFF`	Reserved	64KB
`0x11_0000`	`0x11_FFFF`	Dual Timer SP084	64KB
`0x10_0000`	`0x10_FFFF`	Reserved	64KB

Start Address Offset	End Address Offset	Name	Size
0x0F_0000	0x0F_FFFF	Watchdog SP805	64KB
0x0B_0000	0x0E_FFFF	Reserved	256KB
0x0A_0000	0x0A_FFFF	UART3	64KB
0x09_0000	0x09_FFFF	UART2	64KB
0x08_0000	0x08_FFFF	Reserved	64KB
0x07_0000	0x07_FFFF	KMI 1	64KB
0x06_0000	0x06_FFFF	KMI 0	64KB
0x05_0000	0x05_FFFF	PL180 MCI	64KB
0x03_0000	0x04_FFFF	Reserved	128KB
0x02_0000	0x02_FFFF	SP810 Sysctrl	64KB
0x01_0000	0x01_FFFF	System registers	64KB
0x00_0000	0x00_FFFF	Reserved	64KB

5.3.2 SoC peripherals memory map

The following table summarizes the RoS memory map for the FVP peripherals in terms of offsets from 0x000_7F00_0000 within a 16MB memory space.

Table 5-3: RoS system peripherals memory map

Start Address Offset	End Address Offset	Name	Size
0xFF_0000	0xFF_FFFF	SoC SOR HDLCD security override	64KB
0xFE_0000	0xFE_FFFF	Platform register block	64KB
0xFD_0000	0xFD_FFFF	SMC PL354 Cfg	64KB
0xF9_0000	0xFC_FFFF	Reserved	256KB
0xF8_0000	0xF8_FFFF	UART1	64KB
0xF7_0000	0xF7_FFFF	UART0	64KB
0xF6_0000	0xF6_FFFF	HDLCD0	64KB
0xF5_0000	0xF5_FFFF	HDLCD1	64KB
0xF3_0000	0xF3_FFFF	PCIE Root port	64KB
0xF2_0000	0xF2_FFFF	PCIE Macro	64KB
0xF1_0000	0xF1_FFFF	DMA NS	64KB
0xF0_0000	0xF0_FFFF	DMA S	64KB
0xE9_0200	0xF6_FFFF	Reserved	895KB
0xE9_00FC	0xE9_00FF	I2S ID	3B
0xE9_0000	0xE9_01FF	I2S	512B
0xE8_0000	0xE8_FFFF	Keys	64KB
0xE7_0000	0xE7_FFFF	NV Counter	64KB
0xE6_0000	0xE6_FFFF	TRNG	64KB
0xE5_0000	0xE5_0FFF	surge	4KB
0xE4_0000	0xE4_0FFF	Process, Voltage, and / Temperature (PVT) Monitor apollo	4KB

Start Address Offset	End Address Offset	Name	Size
0xE3_0000	0xE3_0FFF	PVT Monitor atlas	4KB
0xE2_0000	0xE2_0FFF	PVT Monitor gpu	4KB
0xE1_0000	0xE1_0FFF	PVT Monitor soc	4KB
0xE0_0000	0xE0_0FFF	PVT Monitor std	4KB
0xD0_0000	0xDF_FFFF	NIC GPV	1MB
0xB0_0000	0xCF_FFFF	SDRAM PHY 0-31	32*64KB
0x04_0000	0xAF_FFFF	Reserved	11008KB
0x03_0000	0x03_FFFF	SMMU_USB	64KB
0x02_0000	0x02_FFFF	SMMU_HDLCD2	64KB
0x01_0000	0x01_FFFF	SMMU_HDLCD1	64KB
0x00_0000	0x00_FFFF	SMMU_DMA	64KB

5.3.3 Peripherals interrupt map

The following table summarizes the application processor interrupt mappings.

Where:

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

Table 5-4: Application processor interrupt map

INTERRUPT_OFFSET	Source peripheral
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
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

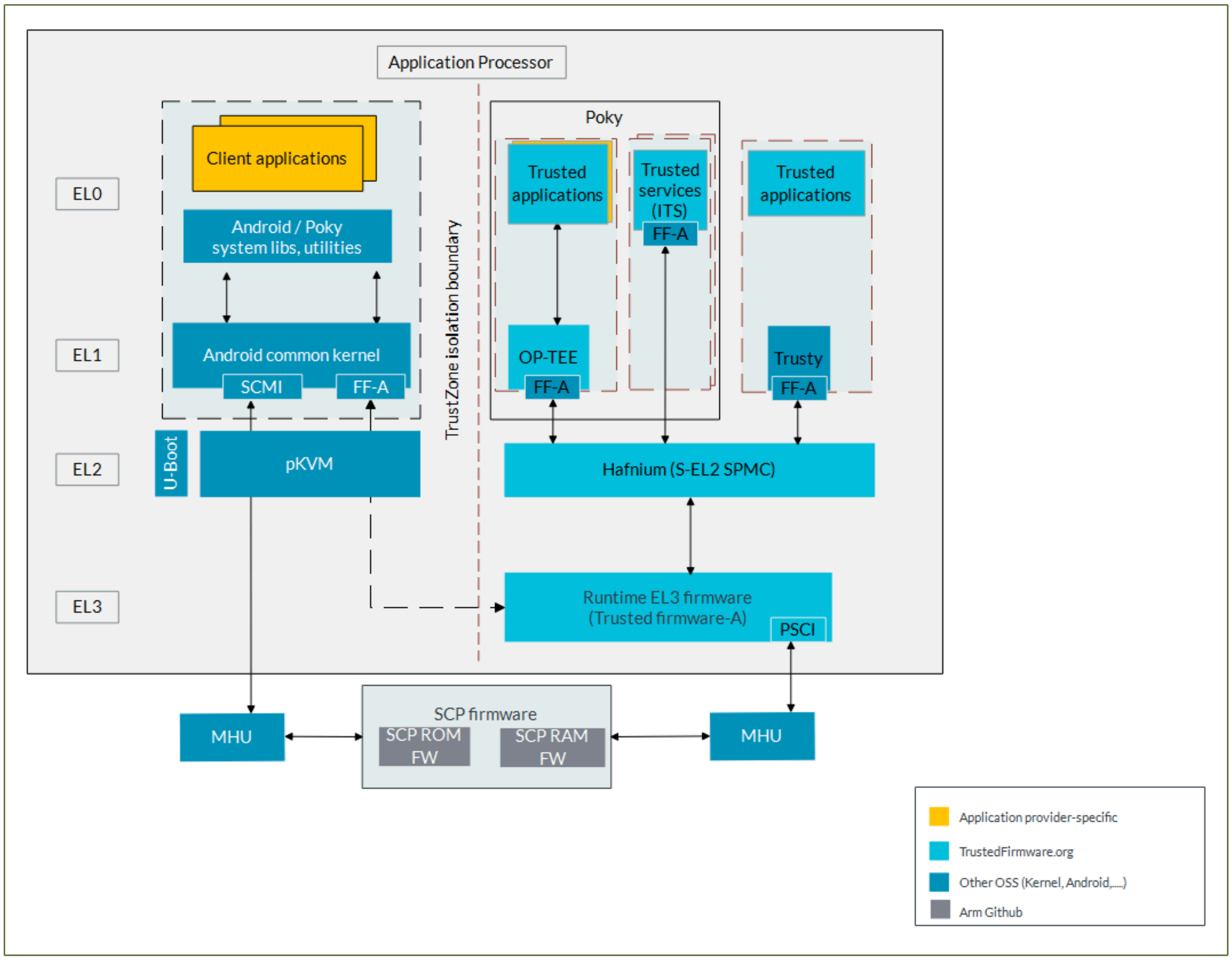
INTERRUPT_OFFSET	Source peripheral
24-28	Reserved
29	HDLCD controller 1
30	SMMU Combined Secure Interrupt
31	SMMU Combined Non-secure Interrupt
32	Reserved
33	Reserved
34-41	DMA0 IRQ7-0
42	DMA0 IRQ ABORT
43	TRNG
44-51	DMA1 IRQ7-0
52	DMA1 IRQ ABORT
53-73	Reserved
74	Virtio Block Device
75	Reserved
76	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. Software

The Total Compute 2021 Software provides a starting point to modify, extend, and develop the Software stack for a *System on Chip* (SoC) based on the Total Compute 2021 Reference Design (RD-TC21).

The following figure shows a runtime view of the Total Compute 2021 Software stack.

Figure 6-1: Runtime view of Total Compute 2021 Software stack



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

6.1 Software components

The Total Compute 2021 Software stack consists of firmware, Kernel, and file system components that can run on the associated *Fixed Virtual Platform* (FVP).

The Software stack has the following components. See Runtime view of Total Compute 2021 Software stack.

SCP firmware, see SCP firmware

Manages the overall power, clock, reset, and system control of the system. For example:

- System initialization
- Clock and power management

AP firmware, see AP firmware

The *Application Processor* (AP) is required to boot the reference design up to the point where the OS execution starts. The AP firmware comprises the following components:

- Arm Trusted firmware BL1
- Arm Trusted firmware BL2
- Arm Trusted firmware BL31
    - *Power State Coordination Interface* (PSCI)
    - Secure Monitor framework
- Secure Partition Manager
- Secure Partitions
    - *Open Portable Trusted Execution Environment* (OP-TEE) Trusted OS
    - Trusted Services with Shim layer

U-Boot or *Unified Extensible Firmware Interface* (UEFI), see U-Boot

Performs hardware initialization and acts as boot loader for rich operating system like Linux.

Kernel, see Kernel

RD-TC21 FVP uses the Android common Kernel. The Linux Kernel contains the RD-TC21 subsystem specific features that demonstrates the capabilities of the FVP platform.

Supports the following Kernel features, that are required to develop and demonstrate the *Compute SubSystem* (CSS) hardware features:

- Mailbox hardware unit
- Supports the following architectural features:
    - *Pointer Authentication Code* (PAC)
    - *Memory Tagging Extension* (MTE)
    - *Branch Target Identification* (BTI)

Android, see Android

RD-TC21 FVP platform runs Android mobile operating system. It supports the following 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 in 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 violation and to increase robustness against attacks that 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.

For more information about the software package and how to set up, see the Arm® Total Compute solution.

6.1.1 SCP firmware

The *System Control Processor* (SCP) firmware is a compute unit of the Total Compute 2021 Reference Design (RD-TC21) subsystem, and is responsible for low-level system management.

The SCP is a Cortex®-M3 processor with a set of dedicated peripherals and interfaces that you can extend.

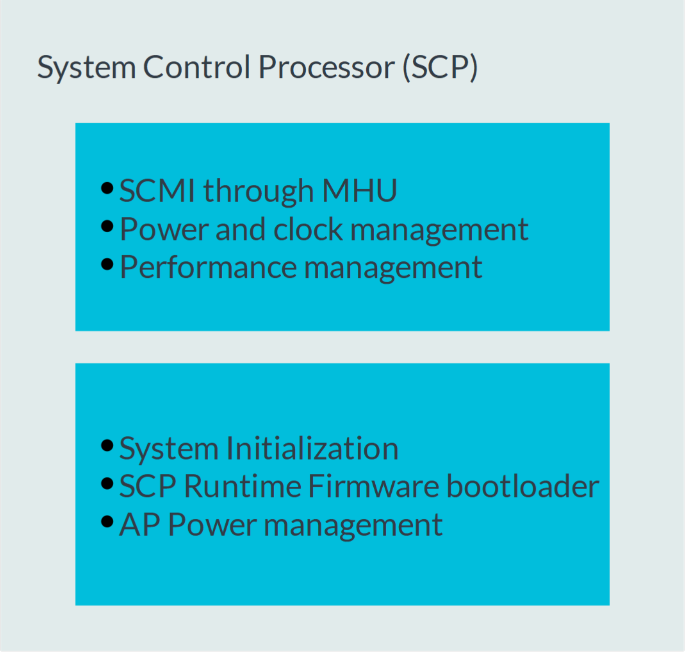
The SCP firmware manages the overall power, clock, reset, and system control of subsystem. The SCP firmware is an inherently Trusted part of the Software. To prevent tampering, internal RAM is used as execution and private storage for all the memory.

SCP firmware supports:

- Powerup sequence and system startup
- Initial hardware configuration
- Clock management
- Servicing power state requests from the *OS Power Management* (OSPM) Software

The following figure shows the SCP firmware components.

Figure 6-2: SCP firmware


System Control Processor (SCP)
• SCMI through MHU
• Power and clock management
• Performance management
• System Initialization
• SCP Runtime Firmware bootloader
• AP Power management

SCP boot ROM

The SCP boot ROM code executes after a Cold reset or power up.

The SCP boot ROM performs the following functions:

- Sets up the generic timer, *Universal Asynchronous Receiver/Transmitter* (UART) console, and clocks
- Initializes the CoreLink™ CI-700
- Powers ON primary *Application Processor* (AP)
- Loads SCP runtime firmware

SCP runtime firmware

The SCP runtime firmware executes after *Trusted Firmware for A-profile* (TF-A) BL2 has authenticated and copied it from flash.

The SCP runtime firmware performs the following functions:

- Responds 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)
- Power domain management
- Clock management

6.1.2 Secure firmware

Secure firmware is a Trusted Software component that runs in the *Application Processor* (AP) Secure world.

It consists of the following:

- AP firmware
- Secure partition manager
- Secure partitions for *Open Portable Trusted Execution Environment* (OP-TEE) and Trusted services

6.1.2.1 AP firmware

The Application Processor (AP) firmware consists of the code that is required to boot Total Compute 2021 Reference Design (RD-TC21) *Fixed Virtual Platform* (FVP) up to the point where the OS execution starts.

This firmware performs architecture and platform initialization. It also loads and initializes Secure world images like Secure partition manager and Trusted OS.

6.1.2.1.1 Arm Trusted firmware BL1

The *Application Processor* (AP) Trusted ROM contains an on-chip Trusted ROM that runs the boot code on Total Compute 2021 Reference Design (RD-TC21) *Fixed Virtual Platform* (FVP).

The AP Trusted firmware BL1 supports the following:

- Minimal architectural initialization like exception vectors and processor initialization
- Platform initialization

It loads the BL2 image and passes control to it.

6.1.2.1.2 Arm Trusted firmware BL2

The *Application Processor* (AP) Trusted firmware BL2 runs at S-EL1. It performs architectural initialization required for subsequent stages of *Trusted Firmware for A-profile* (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 Non-secure use.

The AP Trusted firmware BL2 loads the following images:

- *System Control Processor* (SCP), BL2 image
- EL3 Runtime Software, BL31 image
- Secure Partition Manager, BL32 image
- Non-Trusted firmware - U-boot, BL33 image

- Secure Partitions images, *Open Portable Trusted Execution Environment* (OP-TEE) and Trusted services

6.1.2.1.3 Arm Trusted firmware BL31

The *Application Processor* (AP) Trusted firmware BL2 loads EL3 Runtime Software, BL31. The AP Trusted firmware BL1 passes control to BL31 at EL3.

In Total Compute 2021 Reference Design (RD-TC21) BL31 runs at Trusted SRAM. It provides the following runtime services:

Power State Coordination Interface

The AP Trusted firmware BL31 defines a *Secure Monitor Call* (SMC) interface to support Rich OS Power Management in accordance with the *Power State Coordination Interface* (PSCI) System Software on Arm Systems. 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-TC21 enables SPMD as the Secure payload dispatcher. SPMD relay the *Firmware Framework for A-profile* (FF-A) messages from normal to Secure world or vice-versa. This replaces Trusted OS specific dispatcher from EL3 runtime firmware.

6.1.2.2 Secure partitions

Software image that is isolated using *Secure Partition Manager* (SPM) is Secure partition. RD-TC21 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. OP-TEE OS for Total Compute is built with FFA and SEL2 *Secure Partition Manager Core* (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.

Trusted Services

The PSA cryptography and Secure storage functional APIs are implemented as services in the Trusted Services project managed by Trusted Firmware. They adhere to the FF-A framework specification, and are treated as separate S-EL0 Secure partitions managed by Hafnium at S-EL2 which is acting as the Secure Partition Manager. To manage the S-EL0 applications from S-EL2, the *Virtual Host Extension* (VHE) is enabled in Hafnium.

6.1.2.3 Secure Partition Manager

RD-TC21 enables FEAT S-EL2 architectural extension, and it uses Hafnium as *Secure Partition Manager Core* (SPMC). The BL32 option in *Trusted Firmware for A-profile* (TF-A) is repurposed to specify the SPMC image. The SPMC component runs at S-EL2 exception level.

6.1.3 U-Boot

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

Trusted Firmware for A-profile (TF-A) BL31 passes execution control to U-boot bootloader, BL33. U-boot in RD-TC21 supports the following image formats:

FitImage format

Contains the Linux Kernel and poky ramdisk, which are authenticated and loaded in their respective positions in DRAM, and then execution is handed off to the kernel.

Android boot image

Contains the Linux Kernel and Android ramdisk.

If using *Android Verified Boot* (AVB), the `boot.img` file is loaded from 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.

6.1.4 Kernel

The Linux Kernel in Total Compute 2021 Reference Design (RD-TC21) contains the subsystem-specific features that demonstrate the capabilities of the reference *Fixed Virtual Platform* (FVP).

Apart from default configuration it enables the following:

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

6.1.5 Android

Total Compute 2021 Reference Design (RD-TC21) has support for *Android Open-Source Project* (AOSP), which contains the Android framework, native libraries, Android runtime, and the *Hardware Abstraction Layers* (HALs) for the Android operating system.

The RD-TC21 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.

7. Programmers model

The Programmers model describes the Total Compute 2021 Reference Design (RD-TC21) memory locations and bit descriptions needed for programming software to run on the subsystem hardware.

7.1 Memory maps for RD-TC21

The *System Control Processor* (SCP) has its own memory map that differs from the *Application Processor* (AP) memory map.

7.1.1 AP memory map

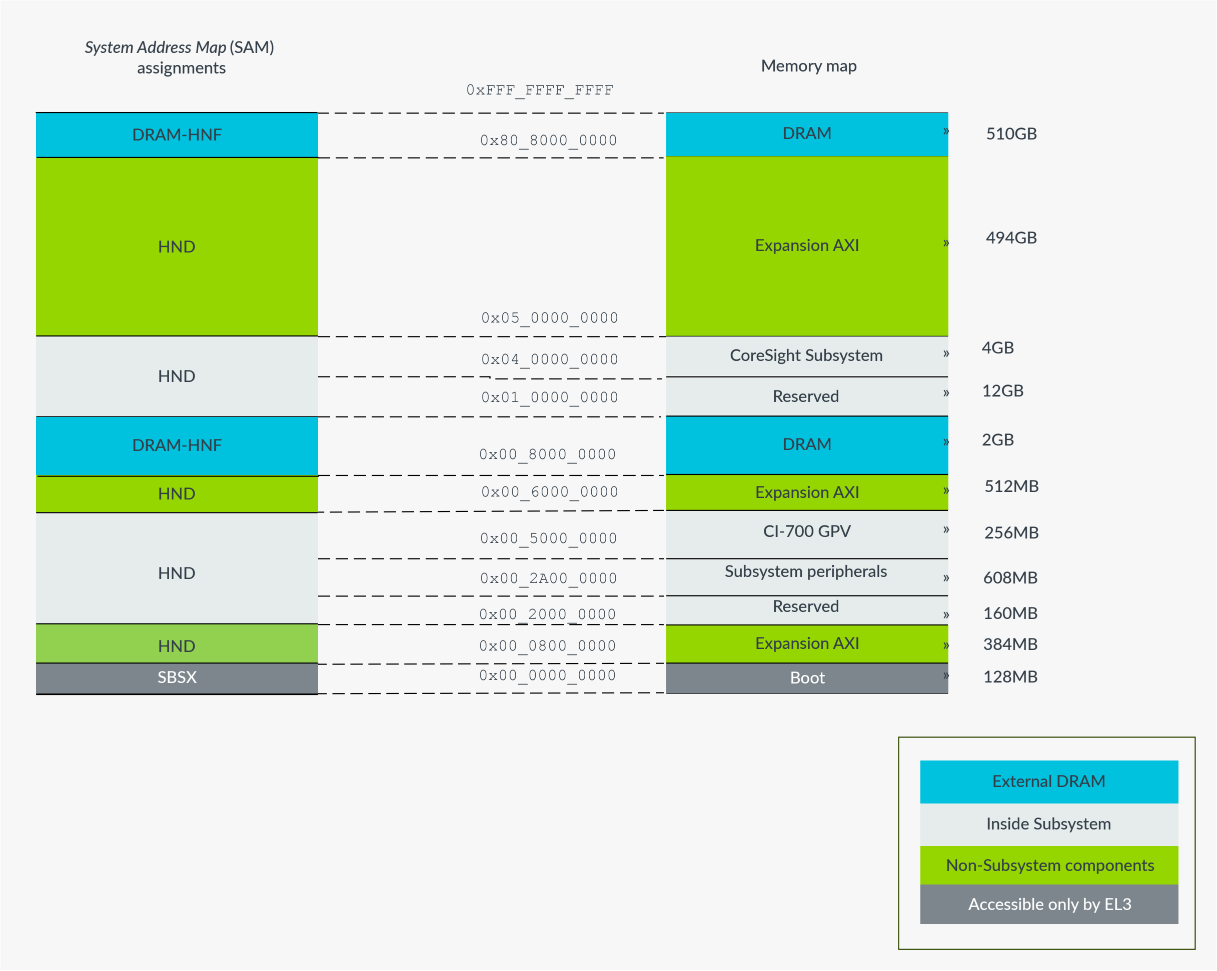
Memory map for the Application Processor.

The *Application Processor* (AP) memory map is visible to the following:

- *Application Processor* (AP)
- *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 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. A region that is defined to be independently software configurable, 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 2021 Reference Design (RD-TC21) 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:

- Where a region maps a peripheral or device. If the peripheral or device occupied less than the region size used, access to the unmapped region results in a DECERR response. For example, 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 themselves determine the response to return. These areas can include unmapped or reserved area 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 lower 2GB of the address space.

There are four separate *Dynamic Memory Controller* (DMC) channels and the following hashing scheme is used to select the DMC channel based on the address bits:

```
stripe_select[1] = Address[9] ^ Address [11] ^ Address [13] ^ Address [27]
^ Address[29] ^ Address [31] ^ Address [33] ^ Address [35] ^ Address [37]
^ Address[39]
```

```
stripe_select[0] = Address [8] ^ Address [10] ^ Address [16] ^ Address [26]
^ Address [28] ^ Address [30] ^ Address [32] ^ Address [34] ^ Address [36]
^ Address [38]
```

Table 7-1: DMC channels and associated address bits

Stripe select	DMC channel
0x0	0
0x1	1
0x2	2
0x3	3

The following table summarizes the memory map for Application Processor.

Table 7-2: AP memory map

Start address	End address	Region	Additional information
0x00_0000_0000	0x00_03FF_FFFF	Secure Boot ROM	Access above 512KB (0x0007_FFFF) will result in a decode error. For accesses above the configured size but below 512KB, reads will return zero

Start address	End address	Region	Additional information
0x00_0400_0000	0x00_04FF_FFFF	Secure RAM	Access 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
0x00_0500_0000	0x00_05FF_FFFF	Non-secure ROM	Access above 512KB (0x0507_FFFF) will result in a decode error. For accesses above the configured size but below 512KB, reads will return zero
0x00_0600_0000	0x00_06FF_FFFF	Non-secure RAM	Access 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
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	-
0x00_2400_0000	0x00_24FF_FFFF	DMC3 configuration	-
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	-
0x00_2800_0000	0x00_28FF_FFFF	TZC400-3 configuration	-
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	-
0x00_2B82_0000	0x00_2CBF_FFFF	Reserved	-
0x00_2CC0_0000	0x00_2CEF_FFFF	Display block	-
0x00_2CF0_0000	0x00_2CFF_FFFF	Reserved	-

Start address	End address	Region	Additional information
`0x00_2D00_0000`	`0x00_2DFF_FFFF`	GPU block	-
`0x00_2E00_0000`	`0x00_2FFF_FFFF`	Reserved	-
`0x00_3000_0000`	`0x00_37FF_FFFF`	GIC-700	-
`0x00_3800_0000`	`0x00_3EFF_FFFF`	Reserved	-
`0x00_3F00_0000`	`0x00_43FF_FFFF`	MMU-700	-
`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 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 *Arm®v7-M Architecture Reference Manual*.

The SCP is a trusted processor and will always run trusted code. Therefore all the regions of SCP memory map that are not mapped to the Application 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, which are generally debug-related components like 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 *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 applications processor 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 applications processor memory map.

This region of the applications processor memory map contains the 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 applications processor 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 applications processor memory map.

This region of the applications processor memory map contains the Boot area, Total Compute 2021 Reference Design (RD-TC21) 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 about memory type 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 Bus Fault exception.

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

Table 7-3: 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	-
`0x4000_0000`	`0x43FF_FFFF`	SCP AHB Expansion	-

Start address	End address	Region	Additional information
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	-
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	-
0xE000_3000	0xE000_DFFF	Reserved	-

Start address	End address	Region	Additional information
`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) – *Advanced Trace Bus* (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 for RD-TC21

This section describes the Total Compute 2021 Reference Design interrupt maps.

7.2.1 AP interrupt map

The *Generic Interrupt Controller* (GIC) Architecture defines two 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.

The following table summarizes the interrupt map of the Application Processor.

Table 7-4: AP interrupt map

Interrupt ID	Interrupt source	Description	Trigger	Polarity
15:0	SGI	Software Generated Interrupt	-	-
16	Reserved	-	-	-
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	-	-	-
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	-	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	Reserved	-	-	-
35:32	DMC0/Reserved	-	-	-
39:36	DMC1/Reserved	-	-	-
43:40	DMC2/Reserved	-	-	-
47:44	DMC3/Reserved	-	-	-
63:48	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
64	Cluster PMU	Cluster PMU Interrupt	-	-
69:65	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
70	CATUADDRERR	From the Debug block	-	-
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

Interrupt ID	Interrupt source	Description	Trigger	Polarity
73	Crypto-REE	From the crypto block	Level	active-HIGH
82:74	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
83	STM-500 synchronization	-	Edge	Rising edge
84	Reserved	-	-	-
85	APP CTI trigger output	--	Edge	Rising edge
86	Trusted Watchdog (WS0)	-	Level	active-HIGH
90:87	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	-	-	-
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

Interrupt ID	Interrupt source	Description	Trigger	Polarity
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
111:110	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
112	DPU TBU PMU interrupt, one per TBU	-	Edge	Rising edge
127:113	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
255:128	Expansion interrupts	**IMPLEMENTATION DEFINED**, as RD-TC21 does not have expansion logic, these will be treated like Reserved interrupts and tied LOW. Blocks like VPU, ISP, 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
283:263	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
304:284	Reserved	-	-	-
347:305	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
475:351	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
476	NCI PMU MESHCLK	-	Level	active-LOW

Interrupt ID	Interrupt source	Description	Trigger	Polarity
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
491:483	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
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
511:499	Reserved	Treated as active-HIGH level interrupt, tied LOW	-	-
512:535	Reserved	-	-	-

7.2.2 SCP interrupt map

Interrupt map for the System Control Processor.

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. *Non-Maskable Interrupt* (NMI) and interrupt IDs 0-31 are wakeup sources.

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

Table 7-5: SCP interrupt map

Interrupt ID	Interrupt source	Description	Trigger	Polarity
NMI	SCP Watchdog	SCP Watchdog	Level	active-HIGH
0	Reserved	Reserved	-	-
1	CoreSight	CoreSight debug power up request	Edge	Rising edge
2	CoreSight	CoreSight system power up request	Edge	Rising edge
3	CoreSight	CoreSight debug reset request	Edge	Rising edge
4:7	Reserved	Reserved	-	-
8	CPU	Cluster Fault/Error IRQ (combined MPAMNSIRQn, MPAMSIRQn CLUSTERCRITIRQn, CLUSTERERRIRQn, and CLUSTERFAULTIRQn)	Level	-
9	CPU	Core Fault/Error IRQ (combined COREERRIRQn, COMPLEXERRIRQn, COREFAULTIRQn, and COMPLEXFAULTIRQn)	Level	-
10:15	Reserved	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	*Generic Interrupt Controller* (GIC) FAULTINT Interrupt. SCP should treat this as a request to reset the whole SYSTOP.	Level	active-HIGH
40	GIC	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	-	-

Interrupt ID	Interrupt source	Description	Trigger	Polarity
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	-	-
50	DMC	DMC2_combined_ecc_err_int	Level	active-HIGH
51	DMC	DMC2_combined_misc_access_int	Level	active-HIGH
52	DMC	TZC400-2 interrupt	Level	active-HIGH
53	DMC	Reserved	-	-
54	DMC	DMC3_combined_ecc_err_int	Level	active-HIGH
55	DMC	DMC3_combined_misc_access_int	Level	active-HIGH
56	DMC	TZC400-3 interrupt	Level	active-HIGH
57:63	Reserved	Reserved	-	-
64	Power Policy Unit	CPU0 Core 0 PPU interrupt	Level	active-HIGH
65	Power Policy Unit	CPU0 Core 1 PPU interrupt	Level	active-HIGH
66	Power Policy Unit	CPU0 Core 2 PPU interrupt	Level	active-HIGH
67	Power Policy Unit	CPU0 Core 3 PPU interrupt	Level	active-HIGH
68	Power Policy Unit	CPU0 Cluster PPU interrupt	Level	active-HIGH
69:73	Reserved	Reserved	-	-
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
77:78	Reserved	Reserved	-	-
79	Power Policy Unit	DPU PPU1 interrupt	Level	active-HIGH
80	Power Policy Unit	DPU PPU2 interrupt	Level	active-HIGH
81	Power Policy Unit	Debug power control logic interrupt	Level	active-HIGH

Interrupt ID	Interrupt source	Description	Trigger	Polarity
82:89	Reserved	Reserved	-	-
90	Power Policy Unit	CPU0 Core 4 PPU interrupt	Level	active-HIGH
91	Power Policy Unit	CPU0 Core 5 PPU interrupt	Level	active-HIGH
92	Power Policy Unit	CPU0 Core 6 PPU interrupt	Level	active-HIGH
94:97	Reserved	Reserved	-	-
98	PLL	Cluster0 *Phase-Locked Loop* (PLL) Lock	Edge	Rising edge
99	Reserved	Reserved	-	-
100	PLL	GPU PLL Lock	Edge	Rising edge
101	Reserved	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
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
112:127	Reserved	Reserved	-	-
128:159	Expansion Interrupts	32 Interrupts for SCP Expansion	-	-

7.3 Register descriptions

Total Compute 2021 Reference Design (RD-TC21) register description summarizes the characterstics of the registers 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 7-6: System ID register summary

Offset	Name	Type	Reset	Width	Description
SID_BASE + 0x0040	SID_SYSTEM_ID	RO	0x00041710	32-bit	ID register
SID_BASE + 0x0050	SID_SOC_ID	RO	0x00000000	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

`0x00041710`

Bit descriptions

Table 7-7: 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 `0x710`, and used for Total Compute 2021 Reference Design (RD-TC21).

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.

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 7-8: 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_NODE_ID register contains information about the node when there are multiple sockets.

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 7-9: SID_NODE_ID bit descriptions

Bits	Name	Description
[31:9]	-	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 the configuration details of the subsystem.

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 7-10: 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 7-11: 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 7-12: 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 7-13: PID1 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (**RAZ**).
[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 7-14: 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 7-15: PID3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Read-As-Zero* (**RAZ**).
[7:0]	-	Reserved, **RAZ**.

7.3.1.10 CID0, System ID Component ID 0 register

The CID0 register contains the first eight bits that specifies segment 0 of the preamble to the system ID component class identifier, 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 + `0x0FF0`

Type

RO

Reset value

`0x0000000D`

Bit descriptions

Table 7-16: 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 segment 1 of the preamble and the system ID component class identifier.

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 7-17: 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 segment 2 of the preamble to the system ID component class identifier.

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 7-18: 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 segment 3 of the preamble to the system ID component class identifier.

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 7-19: 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 2021 Reference Design (RD-TC21) only supports 32-bit word-aligned accesses. Unaligned accesses are treated as **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 7-20: 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 note 1 below	32-bit	Message Handling Unit Configuration
`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
`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

Offset	Name	Type	Reset	Width	Description
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

Note

Note 1: The reset value for MHU_CFG register is equal to the number of channels configured for the particular MHU. See MHU_CFG for more information.

Table 7-21: 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 note 1 below	32-bit	Message Handling Unit Configuration
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
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

Offset	Name	Type	Reset	Width	Description
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

Note

Note 1: The reset value for MHU_CFG register is equal to the number of channels configured for the particular MHU. See MHU_CFG for more information.

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 7-22: 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 7-23: 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 7-24: 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: • Sets a bit that is already set, the bit remains set. • Clears a bit that is already cleared, the bit remains cleared. • Sets and clears a bit at the same time, the bit remains set. We recommend 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 7-25: 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 non-zero, 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 7-26: 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 7-27: 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 7-28: 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 7-29: 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 7-30: 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 7-31: 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 7-32: 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 7-33: 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 7-34: 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 7-35: 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]	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 7-36: 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 7-37: 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 7-38: 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 7-39: 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 7-40: 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 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`

7.3.2.13 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 7-42: 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 7-43: 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 7-44: AIDR bit descriptions

Bits	Name	Description	Type	Default
[31:8]	-	Reserved.	RO	`0x00_0000`

Bits	Name	Description	Type	Default
[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**.	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 7-45: 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 7-46: 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 7-47: 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 7-48: 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 7-49: 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	Customer 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 7-50: 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 7-51: 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 7-52: 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 7-53: 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 CNT control registers

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 CNT Control frame and the REFCLK CNTRead frame in the *Application Processor* (AP) memory map. The base address and accessibility of REFCLK CNT Control frame and the REFCLK CNTRead frame are defined in AP memory 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 summarizes these extra registers in REFCLK CNT control frame and their offset addresses.

Table 7-54: 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 7-55: 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 is used to read back 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 7-56: 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 is used to read back value of the counter sampled on the rising edge of. 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 7-57: 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 2021 Reference Design (RD-TC21) supports various timer frames.

The timer base addresses are mapped in the SCP memory map and AP 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 summarizes the generic timer register summary. Only the PIDn and CIDn registers are described in this section. For more information about the other registers, see the *Arm® Architecture Reference Manual for A-profile architecture*.

Table 7-58: Generic timer register summary

Offset	Name	Type	Reset	Width	Description
0x00	CNTPCT[31:0]	RO	N/A	32-bit	Physical Count register
0x04	CNTPCT[63:32]	RO	N/A	32-bit	Physical Count register
0x08	CNTVCT[31:0]	RO	N/A	32-bit	Virtual Count register
0x0C	CNTVCT[63:32]	RO	N/A	32-bit	Virtual Count register
0x10	CNTFRQ	RW	N/A	32-bit	Counter Frequency register
0x14	CNTPL0ACR	RW	0x0	32-bit	Counter PL0 Access Control register
0x18	CNTVOFF[31:0]	RO	N/A	32-bit	Virtual Offset register
0x1C	CNTVOFF[63:32]	RO	N/A	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 7-59: 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 *Advanced Peripheral Bus* (APB) interface

Table 7-60: 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 7-61: 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 7-62: PID1 bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	Reserved	Reserved, *Read-As-Zero*, *Writes Ignored* (**RAZ/WI**).	RO	`0x0`

Bits	Name	Description	Type	Reset
[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 7-63: 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 0 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 7-64: 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 7-65: 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 7-66: 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 7-67: 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 7-68: 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 Power control registers

Describes the Power control registers.

The *Arm® Power Policy Unit Architecture Specification, version 1.1*, defines the following *Power Policy Unit* (PPU) registers:

- CLUSx-PPU0 Configuration-Dependent registers
- CLUSx-PPU[1-8] Configuration-Dependent registers

See the *Arm® Power Policy Unit Architecture Specification, version 1.1* for more information about PPU registers.

7.3.6 CPU PIK, Core Manager and Power Control registers

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

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

In the following table, it is assumed that the cluster is configured with eight cores of arbitrary type. For example, Cortex®-A715 Core and Cortex®-A510 Core. When the cluster has fewer than eight cores, the registers are assigned to cores of any type contiguously with the higher indices remaining unused.

Table 7-69: 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`	-	DEBUG Clock Divider Control register
`0x0704`	DBGCLK_CLKSEL	RW	`0x00000001`	-	DEBUG Clock Select Control register
0X0708 – `0x0800`	-	RW	-	-	Reserved, **RAZ/WI**

Offset	Name	Type	Reset	Width	Description
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**
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 – 0x097C	CORE<x>CLK configuration registers	RW	-	-	Core Clock Divider and Control registers
0x0980 – 0x09FC	-	RO	-	-	Reserved, **RAZ/WI**
0x0A00 – 0x0A7C	COMPLEX<x>CLK configuration registers	RW	-	-	Complex Core Clock Divider and Control registers
0x0A80 – 0x0BFC	-	RO	-	-	Reserved, **RAZ/WI**
0x0C00	CLKFORCE_STATUS	RO	0x0	32-bit	Clock Force Status register
0x0C04	CLKFORCE_SET	WO	-	32-bit	Clock Force Set register
0x0C08	CLKFORCE_CLR	WO	-	32-bit	Clock Force Clear register
0x0C0C – 0x0DFC	-	RO	-	-	Reserved
0x0E00	ERRIRQ_STATUS	RO	0x0	32-bit	Error *Interrupt Request* (IRQ) Status register
0x0E04	FAULTIRQ_STATUS	RO	0x0	32-bit	Fault IRQ Status register
0x0E08 – 0x0FB0	-	RO	-	32-bit	Reserved, **RAZ/WI**
0x0FB4	CAP3	RO	0x0000_0007	32-bit	Core Capability 3 Definition register
0x0FB8	CAP2	RO	0x0	32-bit	Core Capability 2 Definition register
0x0FBC	CAP	RO	0x70FC_001F	32-bit	Capability Definition register
0x0FC0	PWR_CTRL_CONFIG	RO	0x0014_0000	32-bit	Reserved
0x0FC4 – 0x0FCC	-	RO	-	32-bit	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

Offset	Name	Type	Reset	Width	Description
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 0 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

Describes the Cluster Static Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

0x0000

Type

RW

Reset value

0x0

Bit descriptions

Table 7-70: 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

CPU PIK, Core Manager and Power Control 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 7-71: 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 7-72: 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 7-73: 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0700`

Type

RW

Reset value

`0x0000001F`

Bit descriptions

Table 7-74: 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

Describes the Debug Clock Select Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0704`

Type

RW

Reset value

`0x00000001`

Bit descriptions

Table 7-75: 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

Describes the Clock divider control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0810`

Type

RW

Reset value

`0x0000001F`

Bit descriptions

Table 7-76: 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 registers

Describes the Clock divider control registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0820`

Type

RW

Reset value

`0x0000001F`

Bit descriptions

Table 7-77: 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

Describes the Clock divider control registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

0x0830

Type

RW

Reset value

0x0000001F

Bit descriptions

Table 7-78: 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 registers

Describes the Clock divider control registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

0x0850

Type

RW

Reset value

0x0000001F

Bit descriptions

Table 7-79: 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.

Bits	Name	Description
[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

Describes the CORE<x>CLK and COMPLEX<x>CLK Configuration registers.

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

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

CORE<x>CLK `0x0900 - 0x097C`

COMPLEX<x>CLK `0x0A00 - 0x0A7C`

Type

RW

Bit descriptions

Table 7-80: 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 summarizes the CLKDIV register bit assignment.

Table 7-81: 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.

Bits	Name	Description
[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 summarizes the CLKSEL register bit assignment.

Table 7-82: 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 summarizes the CLKMOD, Clock Modulator register bit assignment.

Table 7-83: 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, Core Manager and Power Control 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0C00`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-84: CLKFORCE_STATUS bit descriptions

Bits	Name	Description
[31:24]	COMPLEX<x>CLKFORCE	One bit per complex clock. Any unused bits are treated as Reserved, *Read-As-Zero, Writes Ignored* (**RAZ/WI**).
[23:16]	-	Reserved, **RAZ/WI**
[15:8]	CORE<x>CLKFORCE	One bit per core clock. Any unused bits are treated as Reserved, **RAZ/WI**.
[7]	-	Reserved, **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, Core Manager and Power Control Clock Force Set register

Describes the Core Manager and Power Control Clock Force Set register.

Writing 1 to a bit in SET disables dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0C04`

Type

WO

Bit descriptions

The bit allocation is the same as the CLKFORCE_STATUS register.

Table 7-85: CLKFORCE_SET bit descriptions

Bits	Name	Description
[31:24]	COMPLEX<x>CLKFORCE	One bit per complex clock. Any unused bits are treated as Reserved (*Read-As-Zero, Writes Ignored* (**RAZ/WI**))
[23:16]	-	Reserved, **RAZ/WI**
[15:8]	CORE<x>CLKFORCE	One bit per core clock. Any unused bits are treated as Reserved (**RAZ/WI**)
[7]	-	Reserved, **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, Core Manager and Power Control Clock Force Clear register

Describes the Core Manager and Power Control Clock Force Clear register.

Writing 1 to a bit in CLR enables dynamic clock gating, writing 0 is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0C08`

Type

WO

Bit descriptions

The bit allocation is the same as the CLKFORCE_STATUS register.

Table 7-86: CLKFORCE_CLR bit descriptions

Bits	Name	Description
[31:24]	COMPLEX<x>CLKFORCE	One bit per complex clock. Any unused bits are treated as Reserved (**RAZ/WI**)
[23:16]	-	Reserved, *Read-As-Zero, Writes Ignored* (**RAZ/WI**)
[15:8]	CORE<x>CLKFORCE	One bit per core clock. Any unused bits are treated as Reserved (**RAZ/WI**)
[7]	-	Reserved, **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 ERRIRQ_STATUS, Error Request Status register

Describes the Error Request Status register.

For all valid ERRIRQn bits:

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0E00`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-87: ERRIRQ_STATUS bit descriptions

Bits	Name	Description
[31:20]	-	Reserved, *Read-As-Zero* (**RAZ**)
[19]	MPAMNSIRQn	MPAMNSIRQ status
[18]	MPAMSIRQn	MPAMSIRQ status
[17]	CLUSTERCRITIRQn	CLUSTERCRITIRQ status
[16:9]	COMPLEXERRIRQn	1 bit per complex. If a complex is not implemented, the associated bit is Reserved, **RAZ**.
[8:1]	COREERRIRQn	1 bit per core. If a core is not implemented, the associated bit is Reserved, **RAZ**.
[0]	CLUSTERERRIRQn	Cluster ERRIRQn status.

7.3.6.14 FAULTIRQ_STATUS, Fault Handling Status register

Describes the Fault Handling Status register.

For all valid FAULTIRQ_STATUS bits:

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0E04`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-88: FAULTIRQ_STATUS register bit assignment

Bits	Name	Description
[31:17]	-	Reserved, *Read-As-Zero* (**RAZ**).
[16:9]	COMPLEXFAULTIRQn	One bit per complex. If a complex is not implemented, the associated bit is Reserved, **RAZ**.
[8:1]	COREFAULTIRQn	One bit per core. If a core is not implemented, the associated bit is Reserved, **RAZ**.
[0]	CLUSTERFAULTIRQn	Cluster FAULTIRQn status.

7.3.6.15 CAP, Core Manager and Power Control Capability Identification register

Describes the Core Manager and Power Control Capability Identification register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FBC`

Type

RO

Reset value

`0x70FC_001F`

Bit descriptions

Table 7-89: CAP bit descriptions

Bits	Name	Description
[31:28]	NUM_PE	Number of processing elements in the cluster. • `0x0` – 1 PE • Up to `0xF` – 16 PEs
[27:24]	-	Reserved
[23]	COMPLEX7CLK_NOT_PRESENT	Indicates whether COMPLEX7CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[22]	COMPLEX6CLK_NOT_PRESENT	Indicates whether COMPLEX6CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[21]	COMPLEX5CLK_NOT_PRESENT	Indicates whether COMPLEX5CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[20]	COMPLEX4CLK_NOT_PRESENT	Indicates whether COMPLEX4CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[19]	COMPLEX3CLK_NOT_PRESENT	Indicates whether COMPLEX3CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.

Bits	Name	Description
[18]	COMPLEX2CLK_NOT_PRESENT	Indicates whether COMPLEX2CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[17]	COMPLEX1CLK_NOT_PRESENT	Indicates whether COMPLEX7CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[16]	COMPLEX0CLK_NOT_PRESENT	Indicates whether COMPLEX7CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[15:9]	-	Reserved
[8]	CORE7CLK_NOT_PRESENT	Indicates whether CORE7CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[7]	CORE6CLK_NOT_PRESENT	Indicates whether CORE6CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[6]	CORE5CLK_NOT_PRESENT	Indicates whether CORE5CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[5]	CORE4CLK_NOT_PRESENT	Indicates whether CORE4CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[4]	CORE3CLK_NOT_PRESENT	Indicates whether CORE3CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[3]	CORE2CLK_NOT_PRESENT	Indicates whether CORE2CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[2]	CORE1CLK_NOT_PRESENT	Indicates whether CORE1CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[1]	CORE0CLK_NOT_PRESENT	Indicates whether CORE0CLK is present or not. • `0x0` – Clock is present. • `0x1` – Clock is not present.
[0]	CLUSSYNC	Indicates whether the Cluster is synchronous to the interconnect or not. • `0x0` – Asynchronous. • `0x1` – Synchronous (includes N:1 sync).

7.3.6.16 CAP2, Core Manager and Power Control Capability Identification 2 register

Describes the Core Manager and Power Control Capability Identification 2 register.

The THREADS_CORE<x> fields have the following mapping:

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FB8`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-90: CAP2 bit descriptions

Bits	Name	Description
[31:16]	-	Reserved
[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

7.3.6.17 CAP3, Core Manager and Power Control Capability Identification 3 register

Describes the Core Manager and Power Control Capability Identification 3 register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FB4`

Type

RO

Reset value

`0x0000_0007`

Bit descriptions

Table 7-91: CAP3 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved
[7]	CPUPLLCLK7_NOT_PRESENT	Indicates whether CPUPLLCLK7 has been implemented or not. `0x0` – *Phase-Locked Loop* (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
[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

7.3.6.18 PWR_CTRL_CONFIG, Core Manager and Power Control Configuration register

Describes the Core Manager and Power Control Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FC0`

Type

RO

Reset value

`0x0014_0000`

Bit descriptions

Table 7-92: PWR_CTRL_CONFIG bit descriptions

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

7.3.6.19 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FD0`

Type

RO

Reset value

`0x0000_0044`

Bit descriptions

Table 7-93: 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`.

7.3.6.20 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**).

7.3.6.21 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FE0`

Type

RO

Reset value

`0x0000_00B8`

Bit descriptions

Table 7-94: 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`

7.3.6.22 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FE4`

Type

RO

Reset value

`0x0000_00B0`

Bit descriptions

Table 7-95: 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`

7.3.6.23 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FE8`

Type

RO

Reset value

`0x0000_0007`

Bit descriptions

Table 7-96: 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.

7.3.6.24 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FEC`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-97: 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`

7.3.6.25 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FF0`

Type

RO

Reset value

`0x0000_000D`

Bit descriptions

Table 7-98: 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`.

7.3.6.26 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FF4`

Type

RO

Reset value

`0x0000_00F0`

Bit descriptions

Table 7-99: 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`

7.3.6.27 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FF8`

Type

RO

Reset value

`0x0000_0005`

Bit descriptions

Table 7-100: 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`

7.3.6.28 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

CPU PIK, Core Manager and Power Control registers

Address offset

`0x0FFC`

Type

RO

Reset value

`0x0000_00B1`

Bit descriptions

Table 7-101: 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 PIK, System Power Integration Kit registers

The System PIK registers enable configuration of clock settings for the system power integration kit.

The System PIK is only mapped into the 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 7-102: 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 7-103: 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**
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) *Advanced Peripheral Bus* (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

Offset	Name	Type	Reset	Width	Description
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**
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

Describes the System PPU Clock Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x800`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-104: 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

Bits	Name	Description
[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

Describes the System PPU Clock Divider Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x804`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-105: 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

Describes the Cache Coherent Interconnect Clock Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x820`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-106: 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

Bits	Name	Description
[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

Describes the Cache Coherent Interconnect Clock Divider Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x824`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-107: 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

Describes the TCU<n> Clock Control register.

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

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

System PIK, System Power Integration Kit registers

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-108: 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

Bits	Name	Description
[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

Describes the TCU<n> Clock Divider Control register.

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

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

System PIK, System Power Integration Kit registers

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-109: 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

Describes the GIC Clock Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x850`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-110: 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
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source: • 0000_0000 – Clock Gated • 0000_0001 – REFCLK • 0000_0010 – SYSPLLCLK • Other values are Reserved

Bits	Name	Description
[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

Describes the GIC Clock Divider Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x854`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-111: 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

Describes the System Peripheral Clock Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x870`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-112: 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

Bits	Name	Description
[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

Describes the System Peripheral Clock Divider Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x874`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-113: 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

Describes the SCP APB Clock Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x860`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-114: 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

Bits	Name	Description
[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

Describes the SCP APB Clock Divider Control register.

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

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x864`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-115: 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

Describes the SCLK Clock Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x880`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-116: 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 DMCCLK_DIV1, SCLK Clock Divider Control register

Describes the SCLK Clock Divider Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x884`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-117: 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

Describes the Non-secure UART Clock Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x8A0`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

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

Table 7-118: 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

Describes the Secure UART Clock Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x8A8`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

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

Table 7-119: 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

Describes the Non-secure UART Clock Divider Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x8A4`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

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

Table 7-120: 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

Describes the Secure UART Clock Divider Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0x8AC`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

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

Table 7-121: 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 PIK, System Power Integration Kit 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 7-122: CLKFORCE_STATUS bit descriptions

Bits	Name	Description
[31:14]	-	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]	-	Reserved
[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 one to a bit within the CLKFORCE_SET register disables any dynamic hardware clock gating, while writing zero to a bit is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0xA04`

Type

WO

Bit descriptions

The bit allocation is the same as the CLKFORCE_STATUS register.

Table 7-123: CLKFORCE_SET bit descriptions

Bits	Name	Description
[31:14]	-	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]	-	Reserved
[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 one to a bit within the CLKFORCE_CLR register disables any dynamic hardware clock gating, while writing zero to a bit is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0xA08`

Type

WO

Reset value

`0x0`

Bit descriptions

The bit allocation is the same as the CLKFORCE_STATUS register.

Table 7-124: CLKFORCE_CLR bit descriptions

Bits	Name	Description
[31:14]	-	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]	-	Reserved
[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

Describes the SYSTOP Reset Delay register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0xB0C`

Type

RW

Reset value

`0x0000_0018`

Bit descriptions

Table 7-125: : 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

Describes the System PIK Power Control Logic Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

System PIK, System Power Integration Kit registers

Address offset

`0xFC0`

Type

RO

Reset value

`0x0021_0002`

Bit descriptions

Table 7-126: PWR_CTRL_CONFIG bit descriptions

Bits	Name	Description
[31:16]	-	ID. This field is set to `0x0024`.
[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 PIK, System Power Integration Kit registers

Address offset

`0xFD0`

Type

RO

Reset value

`0x44`

Bit descriptions

Table 7-127: 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 PIK, System Power Integration Kit registers

Address offset

`0xFE0`

Type

RO

Reset value

`0x00`

Bit descriptions

Table 7-128: 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 PIK, System Power Integration Kit registers

Address offset

`0xFE4`

Type

RO

Reset value

`0xB0`

Bit descriptions

Table 7-129: 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 PIK, System Power Integration Kit registers

Address offset

`0xFE8`

Type

RO

Reset value

`0x0B`

Bit descriptions

Table 7-130: PID2 bit descriptions

Bits	Name	Description
[31:8]	-	Reserved, *Should-Be-Zero* (SBZ).
[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 PIK, System Power Integration Kit registers

Address offset

`0xFEC`

Type

RO

Reset value

`0x00`

Bit descriptions

Table 7-131: 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 PIK, System Power Integration Kit registers

Address offset

`0xFF0`

Type

RO

Reset value

`0x0D`

Bit descriptions

Table 7-132: 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 PIK, System Power Integration Kit registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-133: 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 PIK, System Power Integration Kit registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-134: 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 PIK, System Power Integration Kit registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-135: 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 GPU PIK, GPU Power Control Logic registers

Describes the GPU Power Control Logic registers.

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

Table 7-136: GPU power control logic control 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
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 upon 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
0x1000 – 0x1FFC	GPU-PPU0	-	-	-	*Power Policy Unit* (PPU) Configuration dependent registers
0x2000 – 0x2FFC	-	-	-	-	Reserved, **RAZ/WI**

7.3.8.1 GPUCLK_CTRL, GPU Clock Control register

Describes the GPU Clock Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

`0x810`

Type

`0x0000_0001`

Reset value

RW

Bit descriptions

Table 7-137: 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

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**.

7.3.8.2 GPUCLK_DIV, GPU Clock Divider Control registers

Describes the GPU Clock Divider Control registers.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

`0x814`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-138: 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	The 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

7.3.8.3 GPUCORECLK_CTRL, GPU Clock Control register

Describes the GPU Clock Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

0x0820

Type

RW

Reset value

0x0000_0001

Bit descriptions

Table 7-139: 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**.

7.3.8.4 GPUCORECLK_DIV, GPU Clock Divider Control registers

Describes the GPU Clock Divider Control registers.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

`0x0824`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-140: 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
[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

7.3.8.5 GPUSTACKSCLK_CTRL, GPU Clock Control registers

Describes the GPU Clock Control registers.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

`0x0830`

Type

RW

Reset value

`0x0000_0001`

Bit descriptions

Table 7-141: 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
[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**.

7.3.8.6 GPUSTACKSCLK_DIV, GPU Clock Divider Control registers

Describes the GPU Clock Divider Control registers.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

`0x0834`

Type

RW

Reset value

`0x0000_001F`

Bit descriptions

Table 7-142: 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

7.3.8.7 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 PIK, GPU Power Control Logic 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 7-143: CLKFORCE_STATUS bit descriptions

Bits	Name	Description
[31:4]	-	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.
[1]	GPUCORECLKFORCE	Clock force for GPUCORECLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[0]	GPUCLKFORCE	Clock force for GPUCLK. This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.

7.3.8.8 CLKFORCE_SET, GPU Clock Force SET register

Writing a one to a bit within the CLKFORCE_SET register disabled any dynamic hardware clock gating for that respective clock, while writing zero 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 PIK, GPU Power Control Logic registers

Address offset

`0xA04`

Type

WO

Bit descriptions

Table 7-144: CLKFORCE_SET bit descriptions

Bits	Name	Description
[31:4]	-	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.
[1]	GPUCORECLKFORCE	Clock force for GPUCORECLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[0]	GPUCLKFORCE	Clock force for GPUCLK. This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.

7.3.8.9 CLKFORCE_CLR, GPU Clock Force Clear register

Writing a one to a bit within the CLKFORCE_CLR register enables the dynamic hardware clocking gating for that respective clock, while writing zero 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 PIK, GPU Power Control Logic registers

Address offset

`0xA08`

Type

WO

Reset value

`0x0`

Bit descriptions

Table 7-145: CLKFORCE_CLR bit descriptions

Bits	Name	Description
[31:4]	-	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.
[1]	GPUCORECLKFORCE	Clock force for GPUCORECLK This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.
[0]	GPUCLKFORCE	Clock force for GPUCLK. This field is only available if CAP register bit 0 is set to 1. Otherwise, it is Reserved.

7.3.8.10 CAP, GPU Capabilities register

Describes the GPU Capabilities register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

`0xFBC`

Type

RO

Bit descriptions

Table 7-146: 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.

7.3.8.11 PWR_CTRL_CONFIG, GPU Power Control Logic Configuration register

Describes the GPU Power Control Logic Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

GPU PIK, GPU Power Control Logic registers

Address offset

`0xFC0`

Type

RO

Bit descriptions

Table 7-147: 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).

7.3.8.12 PID4, GPU Peripheral ID4 register

The PID_4 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 PIK, GPU Power Control Logic registers

Address offset

`0xFD0`

Type

RO

Reset value

`0x44`

Bit descriptions

Table 7-148: 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`.

7.3.8.13 PID5-7, GPU Peripheral ID5-7 register

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

7.3.8.14 PID0, GPU Peripheral ID0 register

The PID_0 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 PIK, GPU Power Control Logic registers

Address offset

`0xFE0`

Type

RO

Reset value

`0xB8`

Bit descriptions

Table 7-149: PID0 bit description

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

7.3.8.15 PID1, GPU Peripheral ID1 register

The PID_1 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 PIK, GPU Power Control Logic registers

Address offset

`0xFE4`

Type

RO

Reset value

`0xB0`

Bit descriptions

Table 7-150: 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`

7.3.8.16 PID2, GPU Peripheral ID2 register

The PIDR2 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 PIK, GPU Power Control Logic registers

Address offset

`0xFE8`

Type

RO

Reset value

`0x0B`

Bit descriptions

Table 7-151: 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.

7.3.8.17 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 PIK, GPU Power Control Logic registers

Address offset

`0xFEC`

Type

RO

Reset value

`0x00`

Bit descriptions

Table 7-152: GPU Peripheral ID3 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, 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`

7.3.8.18 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 PIK, GPU Power Control Logic registers

Address offset

`0xFF0`

Type

RO

Reset value

`0x0D`

Bit descriptions

Table 7-153: CID0 bit descriptions

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

7.3.8.19 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 PIK, GPU Power Control Logic registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-154: CID1 bit descriptions

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

7.3.8.20 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 PIK, GPU Power Control Logic registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-155: CID2 bit descriptions

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

7.3.8.21 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 PIK, GPU Power Control Logic registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-156: CID3 bit descriptions

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

7.3.9 Debug PIK, Debug Power Control Logic registers

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

Arm Total Compute 2021 Reference Design (RD-TC21) 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 7-157: 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 7-158: 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
`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

Offset	Name	Type	Reset	Width	Description
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 Advanced Peripheral Bus (APB) Clock Control register
0x824	PCLKDBG_DIV1	RW	0x1F	32-bit	Debug Advanced Peripheral Bus (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**	-	-	-
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

Offset	Name	Type	Reset	Width	Description
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 Chain 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	DBG-PPU	-	-	-	Debug PPU Configuration-dependent registers

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

Describes the Debug Power Request Status 0-3 registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x000`

Type

RO

Reset value

-

Bit descriptions

Table 7-159: 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**.

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

Describes the Debug Power Acknowledge 0-3 registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x010 - 0x01C`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-160: 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

7.3.9.3 DBG_RST_REQ_ST, Debug Reset Request Status register

Describes the Debug Reset Request Status register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x000`

Type

RO

Reset value

-

Bit descriptions

Table 7-161: 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**.

7.3.9.4 DBG_RST_ACK, Debug Reset Acknowledge register

Describes the Debug Reset Acknowledge register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x024`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-162: 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**.

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

Describes the System Power Request Status 0-3 registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x030 – 0x03C`

Type

RO

Reset value

-

Bit descriptions

Table 7-163: 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

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

Describes the System Power Acknowledge 0-3 registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x040 - 0x04C`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-164: 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

7.3.9.7 SYS_RST_REQ_ST, System Reset Request Status register

Describes the System Reset Request Status register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x050`

Type

RO

Reset value

-

Bit descriptions

Table 7-165: 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.

7.3.9.8 SYS_RST_ACK, System Reset Acknowledge register

Describes the System Reset Acknowledge register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x054`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-166: 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**.

7.3.9.9 DBGACK, Debug ACK per PE register

Describes the Debug ACK per PE register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x058`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-167: 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

7.3.9.10 DEBUG_CONFIG, Debug Configuration register

Describes the Debug Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0x008`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-168: DEBUG_CONFIG bit descriptions

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

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

Describes the Debug Clock Control registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

RW

Reset value

See Debug PIK, Debug Power Control Logic registers

Bit descriptions

Table 7-169: 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

Bits	Name	Description
[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**.

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

Describes the Debug Clock Divider Control registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

RW

Reset value

See Debug PIK, Debug Power Control Logic registers

Bit descriptions

Table 7-170: 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

Bits	Name	Description
[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.

7.3.9.13 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 PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

RO

Reset value

See Debug PIK, Debug Power Control Logic 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 7-171: 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

7.3.9.14 CLKFORCE_SET, Debug Clock Force SET register

Writing a one to a bit within the CLKFORCE_SET register disables any dynamic hardware clock gating for that respective clock, while writing zero 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 PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

WO

Reset value

See Debug PIK, Debug Power Control Logic registers

Bit descriptions

Table 7-172: 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

7.3.9.15 CLKFORCE_CLR, Debug Clock Force Clear register

Writing a one to a bit within the CLKFORCE_CLR register enables the dynamic hardware clock gating for that respective clock, while writing zero 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 PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

WO

Reset value

See Debug PIK, Debug Power Control Logic registers

Bit descriptions

Table 7-173: 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

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

Describes the Debug and System Power Request Init Status registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

RW1C

Reset value

See Debug PIK, Debug Power Control Logic registers

Bit descriptions

Table 7-174: 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

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

Describes the Debug and System Reset Request Init Status registers.

Configurations

These registers are available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

RW1C

Reset value

See Debug PIK, Debug Power Control Logic registers

Bit descriptions

Table 7-175: 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**.

7.3.9.18 DEBUG_RCOV, Debug RCOV Control register

Describes the Debug RCOV Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0xB40`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-176: 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

7.3.9.19 CAP, Debug Capabilities register

Describes the Debug Capabilities register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0xFBC`

Type

RO

Reset value

IMPLEMENTATION DEFINED

Bit descriptions

Table 7-177: 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

7.3.9.20 PWR_CTRL_CONFIG, Debug Power Control Logic Configuration register

Describes the Debug Chain Power Control Logic Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

Debug PIK, Debug Power Control Logic registers

Address offset

`0xFC0`

Type

RO

Reset value

`0x00330001`

Bit descriptions

Table 7-178: PWR_CTRL_CONFIG bit descriptions

Bits	Name	Description
[31:16]	-	POWER CONTROL LOGIC_ID. This field is set to `0x0032`.
[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.

7.3.9.21 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 PIK, Debug Power Control Logic registers

Address offset

`0xFD0`

Type

RO

Reset value

`0x44`

Bit descriptions

Table 7-179: 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`.

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

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

7.3.9.23 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 PIK, Debug Power Control Logic registers

Address offset

`0xFE0`

Type

RO

Reset value

`0xB8`

Bit descriptions

Table 7-180: PID0 bit descriptions

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

7.3.9.24 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 PIK, Debug Power Control Logic registers

Address offset

`0xFE4`

Type

RO

Reset value

`0xB0`

Bit descriptions

Table 7-181: 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`

7.3.9.25 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 PIK, Debug Power Control Logic registers

Address offset

`0xFE8`

Type

RO

Reset value

`0x0B`

Bit descriptions

Table 7-182: 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.

7.3.9.26 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 PIK, Debug Power Control Logic registers

Address offset

`0xFEC`

Type

RO

Reset value

`0x00`

Bit descriptions

Table 7-183: 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, 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`

7.3.9.27 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 PIK, Debug Power Control Logic registers

Address offset

`0xFF0`

Type

RO

Reset value

`0x0D`

Bit descriptions

Table 7-184: CID0 bit descriptions

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

7.3.9.28 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 PIK, Debug Power Control Logic registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-185: CID1 bit descriptions

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

7.3.9.29 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 PIK, Debug Power Control Logic registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-186: CID2 bit descriptions

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

7.3.9.30 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 PIK, Debug Power Control Logic registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-187: CID3 bit descriptions

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

7.3.9.31 DEBUG_PPU, Debug PPU Configuration-dependent registers

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 PIK, Debug Power Control Logic registers

Address offset

See Debug PIK, Debug Power Control Logic registers

Type

RW

Reset value

See Debug PIK, Debug Power Control Logic registers

Bit descriptions

Table 7-188: 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`

Bits	Name	Description
[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 7-189: 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

7.3.10 SCP PIK, SCP Power Control Logic registers

The SCP Power Control registers provide access to parameters for the power control logic when configured for a *System Control Processor* (SCP).

The following table summarizes the SCP power control logic register summary.

Table 7-190: 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
0x1000- 0x1FFC	SCP-PPU	-	-	-	*Power Policy Unit* (PPU) Configuration dependent registers
0x2000- 0xFFFC	-	-	-	-	Reserved

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 PIK, SCP Power Control Logic registers

Address offset

`0x010`

Type

RW

Reset value

`0x1`

Bit descriptions

Table 7-191: 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

Describes the WIC based Deep Sleep Control register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP PIK, SCP Power Control Logic registers

Address offset

`0x014`

Type

RW

Reset value

`0x0`

Bit descriptions

Table 7-192: 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

Describes the WIC based Deep Sleep Status register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP PIK, SCP Power Control Logic registers

Address offset

`0x018`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-193: 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 PIK, SCP Power Control Logic 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 7-194: 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 PIK, SCP Power Control Logic registers

Address offset

`0xA04`

Type

WO

Bit descriptions

Writing a one to a bit within the CLKFORCE_SET register disabled any dynamic hardware clock gating for that respective clock, while writing zero to a bit is ignored.

Table 7-195: 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 one to a bit within the CLKFORCE_CLR register enables the dynamic hardware clocking gating for that respective clock, whilst writing zero to a bit is ignored.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP PIK, SCP Power Control Logic registers

Address offset

`0xA08`

Type

WO

Reset value

`0x0`

Bit descriptions

Table 7-196: 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

Describes the PLL Status 0 register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP PIK, SCP Power Control Logic registers

Address offset

`0xA10`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-197: 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]	SYSPLLLOCK_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]	-	Reserved
[1]	-	Reserved
[0]	REFCLK_ST	Status of REFCLK

7.3.10.8 PLL_STATUS1, PLL Status 1 register

Describes the PLL Status 1 register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP PIK, SCP Power Control Logic registers

Address offset

`0xA14`

Type

RO

Reset value

`0x0`

Bit descriptions

Table 7-198: PLL_STATUS1 bit descriptions

Bits	Name	Description
[31:16]	-	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

Bits	Name	Description
[3]	CPU0PLL3 LOCK _ST	Status of CPU0 PLL3
[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

Describes the SCP Power Control Logic Configuration register.

Configurations

This register is available in all configurations.

Attributes

Width

32-bit

Functional group

SCP PIK, SCP Power Control Logic registers

Address offset

`0xFC0`

Type

RO

Reset value

`0x0070_0001`

Bit descriptions

Table 7-199: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFD0`

Type

RO

Reset value

`0x44`

Bit descriptions

Table 7-200: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFE0`

Type

RO

Reset value

`0xB8`

Bit descriptions

Table 7-201: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFE4`

Type

RO

Reset value

`0xB0`

Bit descriptions

Table 7-202: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFE8`

Type

RO

Reset value

`0x0B`

Bit descriptions

Table 7-203: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFEC`

Type

RO

Reset value

`0x00`

Bit descriptions

Table 7-204: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFF0`

Type

RO

Reset value

`0x0D`

Bit descriptions

Table 7-205: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFF4`

Type

RO

Reset value

`0xF0`

Bit descriptions

Table 7-206: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFF8`

Type

RO

Reset value

`0x05`

Bit descriptions

Table 7-207: 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 PIK, SCP Power Control Logic registers

Address offset

`0xFFC`

Type

RO

Reset value

`0xB1`

Bit descriptions

Table 7-208: 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	-