AM5K2E0x Multicore ARM KeyStone II System-on-Chip (SoC)

1 AM5K2E0x Features and Description

1.1 Features

- ARM® Cortex®-A15 MPCore™ CorePac
  - Up to Four ARM Cortex-A15 Processor Cores at up to 1.4-GHz
  - 4MB L2 Cache Memory Shared by all Cortex-A15 Processor Cores
  - Full Implementation of ARMv7-A Architecture Instruction Set
  - 32KB L1 Instruction and Data Caches per Core
  - AMBA 4.0 AXI Coherency Extension (ACE) Master Port, Connected to MSMC (Multicore Shared Memory Controller) for Low Latency Access to SRAM and DDR3
- Multicore Shared Memory Controller (MSMC)
  - 2 MB SRAM Memory for ARM CorePac
  - Memory Protection Unit for Both SRAM and DDR3_EMIF
- Multicore Navigator
  - 8k Multi-Purpose Hardware Queues with Queue Manager
  - One Packet-Based DMA Engine for Zero-Overhead Transfers
- Network Coprocessor
  - Packet Accelerator Enables Support for
    - Transport Plane IPsec, GTP-U, SCTP, PDCP
    - L2 User Plane PDCP (RoHC, Air Ciphering)
    - 1 Gbps Wire Speed Throughput at 1.5 MPackets Per Second
  - Security Accelerator Engine Enables Support for
    - IPSec, SRTP, 3GPP and WiMAX Air Interface, and SSL/TLS Security
    - ECB, CBC, CTR, F8, A5/3, CCM, GCM, HMAC, CMAC, GMAC, AES, DES, 3DES, Kasumi, SNOW 3G, SHA-1, SHA-2 (256-bit Hash), MD5
    - Up to 6.4 Gbps IPSec and 3 Gbps Air Ciphering
  - Ethernet Subsystem
    - Eight SGMII Ports with Wire Rate Switching
    - IEEE1588 v2 (with Annex D/E/F) Support
    - 8 Gbps Total Ingress/Egress Ethernet BW from Core
    - Audio/Video Bridging (802.1Qav/D6.0)
    - QOS Capability
    - DSCP Priority Mapping
- Peripherals
  - Two PCIe Gen2 Controllers with Support for
    - Two Lanes per Controller
    - Supports Up to 5 GBaud
  - One HyperLink
    - Supports Connections to Other KeyStone Architecture Devices Providing Resource Scalability
    - Supports Up to 50 GBaud
  - 10-Gigabit Ethernet (10-GbE) Switch Subsystem
    - Two SGMII/XFI Ports with Wire Rate Switching and MACSEC Support
    - IEEE1588 v2 (with Annex D/E/F) Support
  - One 72-Bit DDR3/DDR3L Interface with Speeds Up to 1600 MTPS in DDR3 Mode
  - EMIF16 Interface
  - Two USB 2.0/3.0 Controllers
  - USIM Interface
  - Two UART Interfaces
  - Three I^2C Interfaces
  - 32 GPIO Pins
  - Three SPI Interfaces
  - One TSIP
    - Support 1024 DS0s
    - Support 2 Lanes at 32.768/16.3848.192 Mbps Per Lane
- System Resources
  - Three On-Chip PLLs
  - SmartReflex Automatic Voltage Scaling
  - Semaphore Module
  - Twelve 64-Bit Timers
  - Five Enhanced Direct Memory Access (EDMA) Modules
- Commercial Case Temperature:
  - 0ºC to 85ºC
- Extended Case Temperature:
  - -40ºC to 100ºC

1.2 Applications

- Avionics and Defense
- Communications
- Industrial Automation
- Automation and Process Control
- Servers
- Enterprise Networking
- Cloud Infrastructure

1.3 KeyStone II Architecture

TI's KeyStone II Multicore Architecture provides a unified platform for integrating RISC processing cores along with both hardware/firmware based application-specific acceleration and high performance I/Os. The KeyStone II Multicore Architecture is a proven device architecture to achieve the full performance entitlement through the following major components: TeraNet, Multicore Shared Memory Controller, Multicore Navigator, and HyperLink.

TeraNet is a multipoint to multipoint non-blocking switch fabric. Its distributed arbiter provides multiple duplex communication channels in parallel between the master and slave ports without interference. The priority based arbitration mechanism ensures the delivery of the critical traffic delivery in the system.

The Multicore Shared Memory Controller (MSMC) is the center of the KeyStone II memory architecture. It provides multiple fast and high-bandwidth channels for processor cores to access DDR and minimizes the access latency by directly connecting to the DDR. The MSMC also provides the flexibility to expand processor cores with little impact at the device level. In addition, it provides multi-bank based fast on-chip SRAM shared among processor cores and IOs. It also provides the I/O cache coherency for the device when the Cortex-A15 processor core is integrated.

The Multicore Navigator provides a packet-based IPC mechanism among processing cores and packet based peripherals. The hardware-managed queues supports multiple-in-multiple-out mode without using mutex. Coupled with the packet-based DMA, the Multicore Navigator provides a highly efficient and software-friendly tool to offload the processing core to achieve other critical tasks.

HyperLink provides a 50-GBaud chip-level interconnect that allows devices to work in tandem. Its low latency, low overhead and high throughput makes it an ideal interface for chip-to-chip interconnections.

There are two generations of KeyStone architecture. The AM5K2E0x device is based on KeyStone II, which integrates a Cortex-A15 processor CorePac.

1.4 Device Description

The AM5K2E0x is a high performance device based on TI's KeyStone II Multicore SoC Architecture, incorporating the most performance-optimized Cortex-A15 processor dual-core or quad-core CorePac that can run at a core speed of up to 1.4 GHz. TI's AM5K2E0x device enables a high performance, power-efficient and easy to use platform for developers of a broad range of applications such as enterprise grade networking end equipment, data center networking, avionics and defense, medical imaging, test and automation.

TI's KeyStone II Architecture provides a programmable platform integrating various subsystems (for example, ARM CorePac (Cortex-A15 Processor Quad Core CorePac), network processing, and uses a queue-based communication system that allows the device resources to operate efficiently and seamlessly. This unique device architecture also includes a TeraNet switch that enables the wide mix of system elements, from programmable cores to high-speed IO, to each operate at maximum efficiency with no blocking or stalling.

The AM5K2E0x KeyStone II device integrates a large amount of on-chip memory. The Cortex-A15 processor cores each have 32KB of L1Data and 32KB of L1 Instruction cache. The up to four Cortex A15 cores in the ARM CorePac share a 4MB L2 Cache. The device also integrates 2MB of Multicore Shared Memory (MSMC) that can be used as a shared L3 SRAM. All L2 and MSMC memories incorporate error detection and error correction. For fast access to external memory, this device includes a 64-bit DDR-3 (72-bit with ECC support) external memory interface (EMIF) running at 1600 MTPS.

The device enables developers to use a variety of development and debugging tools that include GNU GCC, GDB, Open source Linux, Eclipse based debugging environment enabling kernel and user space debugging using a variety of Eclipse plug-ins including TI's industry leading IDE Code Composer Studio.

1.5 Enhancements in KeyStone II

The KeyStone II architecture provides many major enhancements over the previous KeyStone I generation of devices. The KeyStone II architecture integrates an ARM Cortex-A15 processor quad-core cluster to enable Layer 2 (MAC/RLC) and higher layer processing. The external memory bandwidth has been doubled with the integration of dual DDR3 1600 EMIFs. MSMC internal memory bandwidth is quadrupled with MSMC V2 architecture improvements. Multicore Navigator supports 2× the number of queues, descriptors and packet DMA, 4× the number of micro RISC engines and a significant increase in the number of push/pops per second, compared to the previous generation. The new peripherals that have been added include the USB 3.0 controller and Asynchronous EMIF controller for NAND/NOR memory access. The 2-port Gigabit Ethernet switch in KeyStone I has been replaced with an 8-port Gigabit Ethernet switch and a 10 GbE switch in KeyStone II. Time synchronization support has been enhanced to reduce software workload and support additional standards like IEEE1588 Annex D/E and SyncE. The number of GPIOs and serial interface peripherals like I^2C and SPI have been increased to enable more board level control functionality.

1.6 Functional Block Diagram

The figures below show the functional block diagrams of the AM5K2E0x devices.

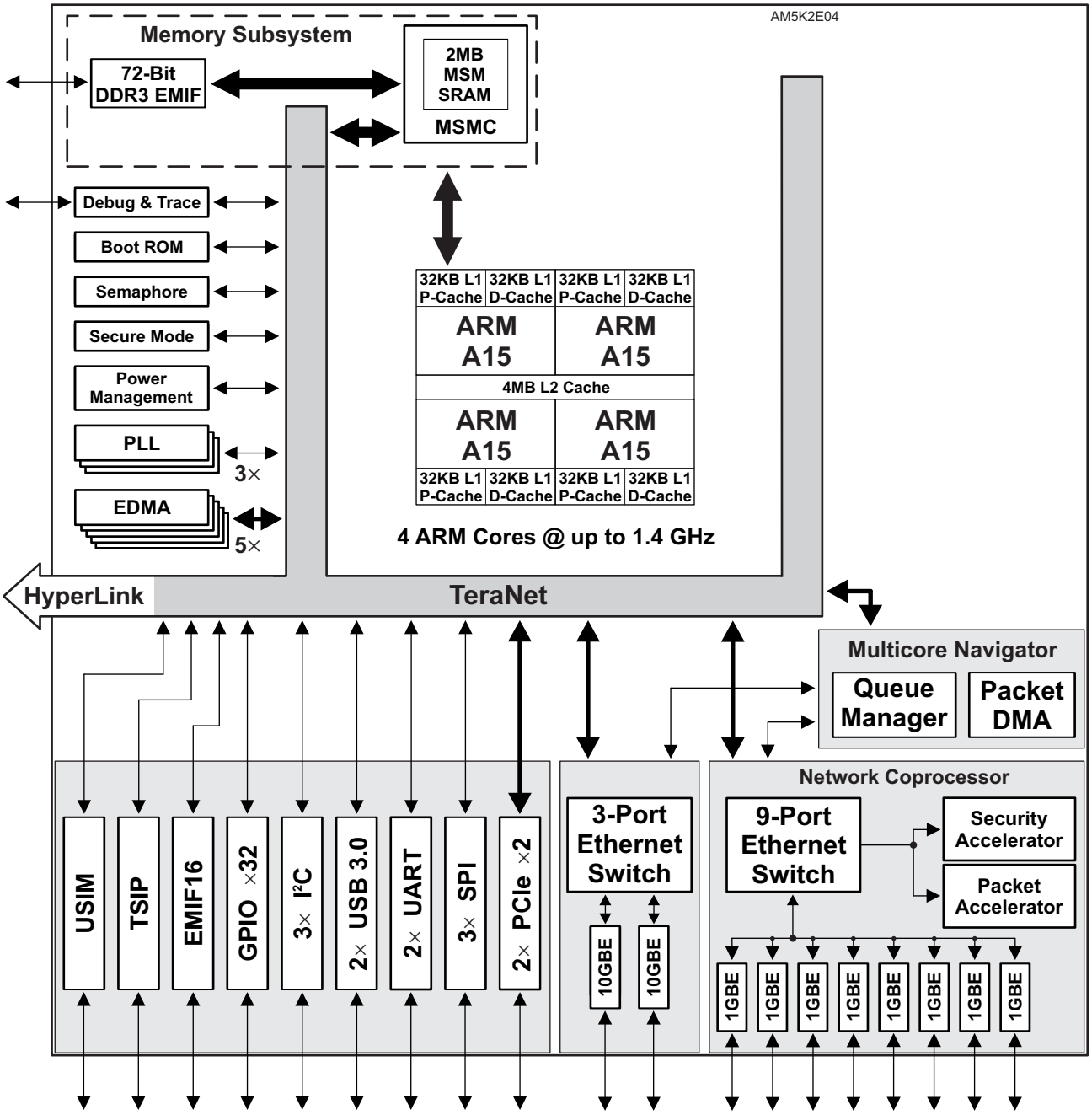


Figure 1-1. AM5K2E04 Functional Block Diagram

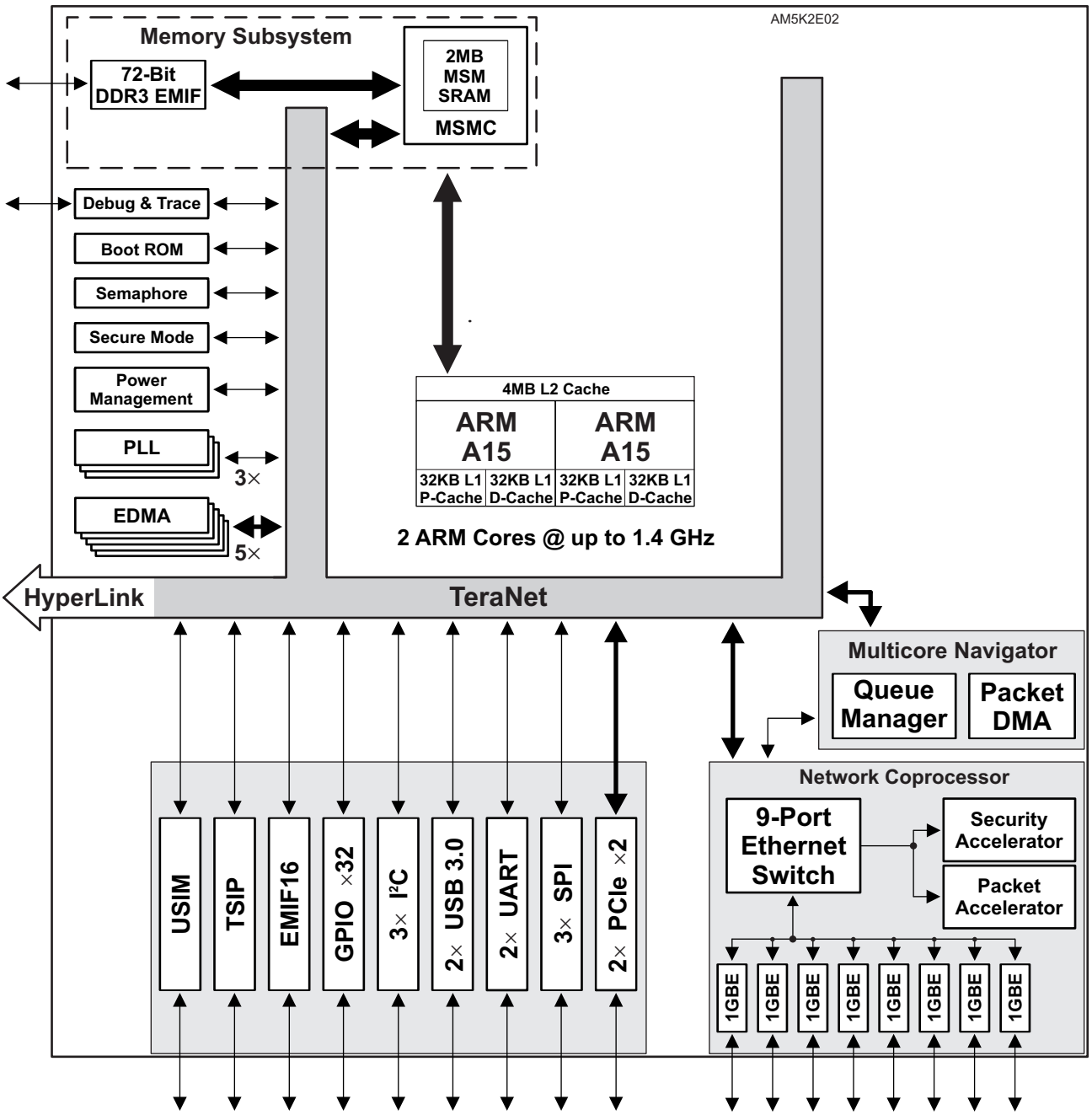


Figure 1-2. AM5K2E02 Functional Block Diagram

Table of Contents

2 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision C (August 2014) to Revision D	Page
• Added Top Navigation links to front page of the document	1
• Changed Product Status to Production Data	1
• Changed Mission Critical Systems to Avionics and Defense in Section 1.2	2
• Changed mission critical to avionics and defense in first paragraph of Section 3.2.1	2
• Changed Product Status to PD and changed footnote (3) in Table 3-1	8
• Changed second list item under Software Development Tools in Section 3.2.1	10
• Added Related Links, Community Resources, Trademarks, Electrostatic Discharge Caution, and Glossary sections to Section 3	13
• Added Figure 4-1	14
• Changed DDR3A to DDR3 in Table 4-1	16
• Changed All instances of DDR3A to DDR3 in Table 5-2	25
• Changed Supply DDR3AREFSSTL to DDR3REFSSTL in Table 5-3	38
• Changed the DVDD15 Volts and Supply Description in Table 5-3	38
• Changed Start Address for PCIe1SerDes Config to 00 0232 6000, End Address for USB 0 MMR CFG to 00 026F FFFF, and all instances of DDR3A to DDR3 in Table 6-1	55
• Changed CPT_DDR3A to CPT_DDR3 in Table 6-6	66
• Changed DDR3A to DDR3 in Event No. 388 Name and Description in Table 6-22	78
• Changed DDR3A to DDR3 in Section 6.4	104
• Changed DDR3A to DDR3 in Section 7	114
• Changed DDR3A to DDR3 in Figure 7-3	117
• Changed DDR3A to DDR3 in Figure 7-6	122
• Added EMIF and NAND to Description in Table 8-2	131
• Changed DDR3A to DDR3 in Section 8.1.4	147
• Changed DDR3APLLCTL0 and DDR3APLLCTL1 to DDR3PLLCTL0 and DDR3PLLCTL1 in Table 8-26	149
• Changed AVSIFSEL Description value 11 to Reserved in Table 8-27	153
• Changed ARMENDIAN_CFG4_0 Default Value to 0x00023A00 in Table 8-42	164
• Changed ARMENDIAN_CFG5_1 Default Value to 0x00000006 in Table 8-44	165
• Changed DDR3AVREFSSTL to DDR3VREFSSTL and DDR3A to DDR3 in Section 9.1	175
• Changed MIN, NOM, and MAX values for CVDD Initial and CVDD1; changed DVDD15 to DDR3 I/O voltage and added values; changed DDR3A to DDR3 and DDR3AVREFSSTL to DDR3VREFSSTL; changed DSP to SOC in footnote (4) in Section 9.2	176
• Changed DDR3A to DDR3 in Section 9.3	177
• Changed DDR3A to DDR3 and changed DVDD15 to DDR3 memory I/O voltage and DDR3 (1.5/1.35 V) I/O Buffer Type in Table 9-1	178
• Changed DDR3A to DDR3 and added 1.35 V to Voltage for DVDD15 in Table 10-1	179
• Changed EMIF(DDR3A) to EMIF(DDR3) in Table 10-6	187
• Changed DDR3A EMIF to DDR3 EMIF in Table 10-7	188
• Changed DDR3A in Section 10.4.3	195
• Changed DDR3A in Section 10.5	198
• Changed Figure 10-7	199
• Deleted second sentence from Section 10.5.1.1	200
• Changed DDR3A to DDR3 in Table 10-13	201
• Changed Address Range 00 0231 0128 to Reserved in Table 10-15	202
• Changed OUTPUT DIVIDE Field Description in Table 10-16	203
• Changed MAX value for tj(CORECLKN) and tj(CORECLKP) in Table 10-27	209
• Changed Figure 10-26	214
• Changed PAPLL Field Description in Table 10-32	215
• Changed MAX value for tc(NETCPCLKN) and tc(NETCPCLKP) in Table 10-33	215
• Changed DDR3A Memory Controller to DDR3 Memory Controller in Section 10.8	216
• Changed MIN and MAX values for t_c(CEL) in Table 10-56	236
• Changed DDR3A to DDR3 in Table 10-62	244

3 Device Characteristics

Table 3-1 provides an overview of the AM5K2E0x device. The table shows the significant features of the device, including the capacity of on-chip RAM, the peripherals, the CPU frequency, and the package type with pin count.

Table 3-1. Characteristics of the AM5K2E0x Processor

HARDWARE FEATURES		AM5K2E02	AM5K2E04
ARM Cores	ARM Cortex A15 Cores	2	4
	ARM L1 instruction cache memory size (per core)	32KB	
	ARM L1 data cache memory size (per core)	32KB	
	ARM L2 unified cache memory size (shared by all cores)	4MB	
Peripherals	DDR3 memory controller (72-bit bus width) [1.5 V/1.35V] (clock source = DDRREFCLKN|P)	1	
	EDMA3 (64 independent channels) [CPU/3 clock rate]	5	
	Hyperlink	1	
	USB 3.0	2	
	USIM(1)	1	
	I^2C	3	
	SPI	3	
	PCIe (2 lanes per instance)	2	
	UART	2	
	10/100/1000/10000 Ethernet ports	0	2
	10/100/1000 Ethernet ports	8	8
	Management Data Input/Output (MDIO)	3	
	64-bit timers (configurable)	Twelve 64-bit ***or*** Twenty four 32-bit	
	General-Purpose Input/Output port (GPIO)	32	
	TSIP	1	
Accelerators	Packet Accelerator	1	
	Security Accelerator(2)	1	
On-Chip L3 Memory	Organization	2MB MSM SRAM 256 KB L3 ROM	
JTAG BSDL_ID	JTAGID Register (address location: 0x02620018)	0x0B9A_602F	
Frequency	ARM-A15 Processor	1.25 GHz 1.4 GHz	
Voltage	Core (V)	SmartReflex variable supply	
	I/O (V)	1.35 V, 1.5 V, 1.8 V, and 3.3 V	
BGA Package	27 mm x 27 mm	1089-Pin Flip-Chip Plastic BGA (ABD)	
Process Technology	nm	28 nm	
Product Status(3)	Product Preview (PP), Advance Information (AI), or Production Data (PD)	PD	

(1) The USIM is implemented for support of secure devices only. Contact your local technical sales representative for further details.
(2) The Security Accelerator function is subject to export control and will be enabled *only* for approved device shipments.
(3) PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

3.1 ARM CorePac

The ARM CorePac of the AM5K2E0x integrates a Cortex-A15 Cluster (4 Cortex-A15 processors) with additional logic for bus protocol conversion, emulation, interrupt handling, and debug related enhancements. The Cortex-A15 processor is an ARMv7A-compatible, multi-issue out-of-order, superscalar pipeline with integrated L1 caches. The implementation also supports advanced SIMDV2 (Neon technology) and VFPv4 (Vector Floating Point) architecture extensions, security, virtualization, LPAE (Large Physical Address Extension), and multiprocessing extensions. The quad core cluster includes a 4MB L2 cache and support for AMBA4 AXI and AXI Coherence Extension (ACE) protocols. For more information see the *KeyStone II Architecture ARM CorePac User's Guide User Guide* (SPRUHJ4).

3.2 Development Tools

3.2.1 Development Support

In case the customer would like to develop their own features and software on the AM5K2E0x device, TI offers an extensive line of development tools for the KeyStone II platform, including tools to evaluate the performance of the processors, generate code, develop algorithm implementations, and fully integrate and debug software and hardware modules. The tool's support documentation is electronically available within the Code Composer Studio™ Integrated Development Environment (IDE).

The following products support development of KeyStone devices:

- **Software Development Tools**:
  - Code Composer Studio Integrated Development Environment (IDE), including Editor C/C++/Assembly Code Generation, and Debug plus additional development tools
  - Scalable, Real-Time foundation software, which provides the basic run-time target software needed to support any application
- **Hardware Development Tools**:
  - Extended Development System (XDS™) Emulator (supports multiprocessor system debug)
  - EVM (Evaluation Module)

3.3 Device Nomenclature

To designate the stages in the product development cycle, TI assigns prefixes to the part numbers of all devices and support tools. Each family member has one of two prefixes: X or [blank]. These prefixes represent evolutionary stages of product development from engineering prototypes through fully qualified production devices/tools.

3.3.1 Device Development Evolutionary Flow

The device development evolutionary flow is as follows:

- **X:** Experimental device that is not necessarily representative of the final device's electrical specifications
- **[Blank]:** Fully qualified production device

Support tool development evolutionary flow:

- **X:** Development-support product that has not yet completed Texas Instruments internal qualification testing.
- **[Blank]:** Fully qualified development-support product

Experimental (X) and fully qualified [Blank] devices and development-support tools are shipped with the following disclaimer:

Developmental product is intended for internal evaluation purposes.

Fully qualified and production devices and development-support tools have been characterized fully, and the quality and reliability of the device have been demonstrated fully. TI's standard warranty applies.

Predictions show that experimental devices (X) have a greater failure rate than the standard production devices. Texas Instruments recommends that these devices not be used in any production system because their expected end-use failure rate still is undefined. Only qualified production devices are to be used.

TI device nomenclature also includes a suffix with the device family name. This suffix indicates the package type (for example, ABD), the temperature range (for example, blank is the default case temperature range), and the device speed range, in Megahertz (for example, blank is 1000 MHz [1 GHz]).

For device part numbers and further ordering information for AM5K2E0x in the ABD package type, see the TI website www.ti.com or contact your TI sales representative.

3.3.2 Part Number Legend

The following figures provide a legend for reading the complete device name for a KeyStone II device.

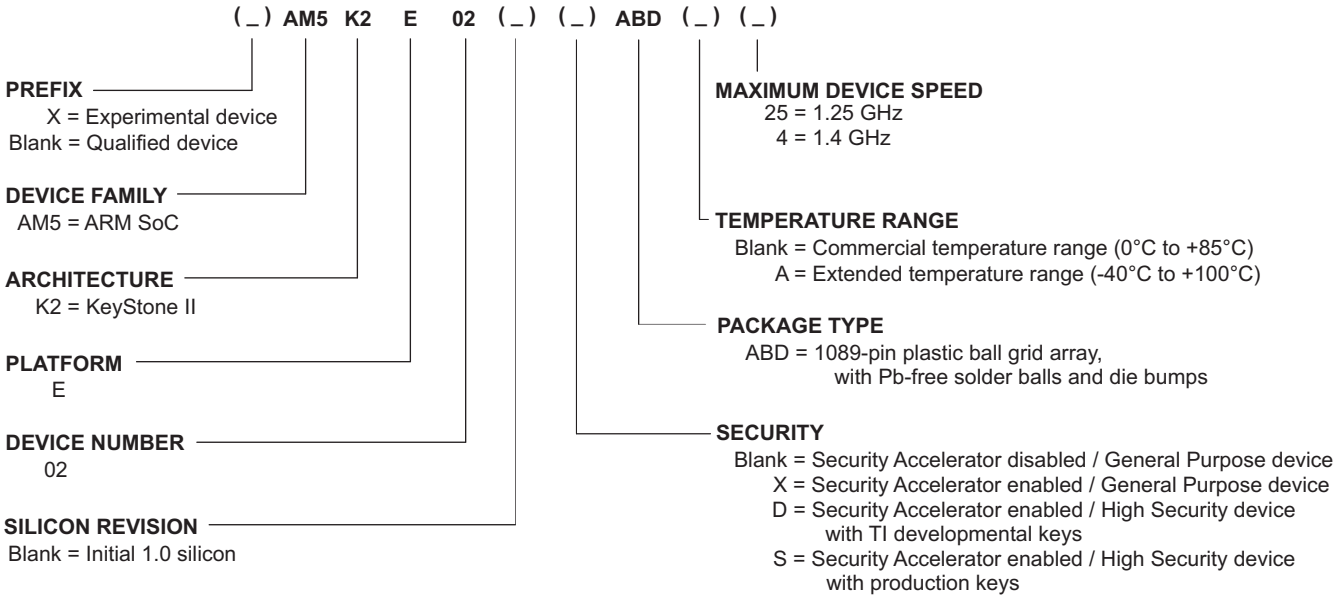


Figure 3-1. Device Nomenclature for AM5K2E02

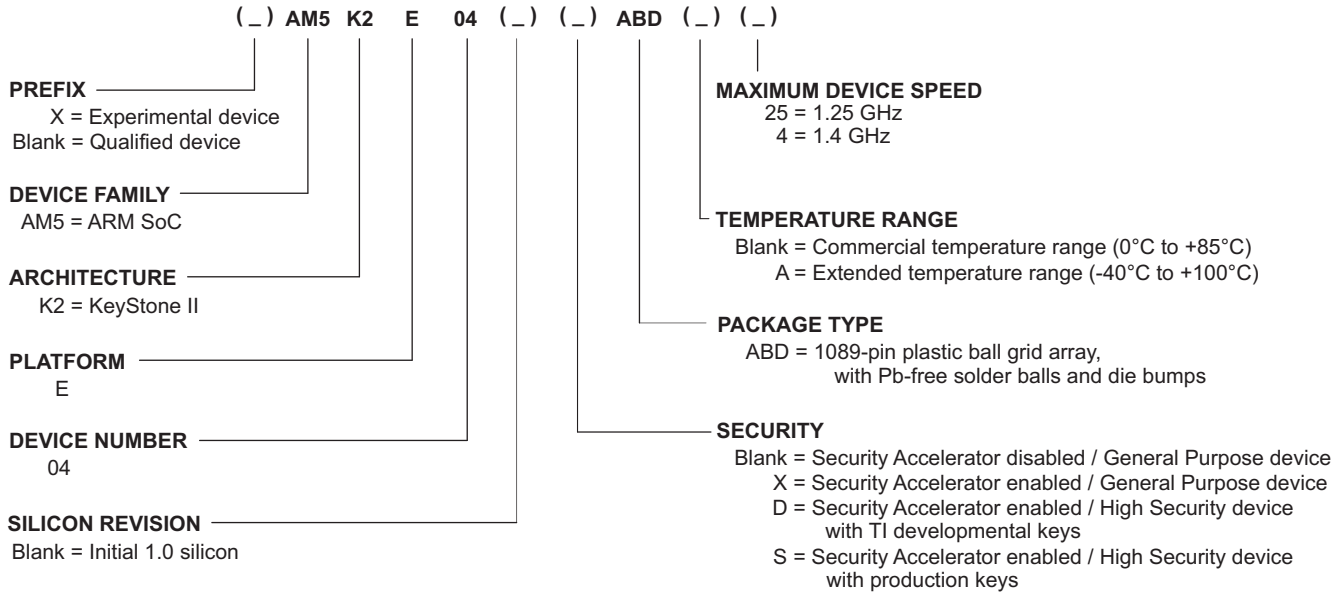


Figure 3-2. Device Nomenclature for AM5K2E04

3.4 Related Documentation from Texas Instruments

These documents describe the AM5K2E0x Multicore ARM KeyStone II System-on-Chip (SoC). Copies of these documents are available on the Internet at www.ti.com.

KeyStone Architecture Timer 64P User's Guide	SPRUGV5
KeyStone II Architecture ARM Bootloader User's Guide	SPRUHJ3
KeyStone II Architecture ARM CorePac User's Guide	SPRUHJ4
KeyStone Architecture Chip Interrupt Controller (CIC) User's Guide	SPRUGW4
KeyStone I Architecture Debug and Trace User's Guide	SPRUGZ2
DDR3 Design Requirements for KeyStone Devices application report	SPRABI1
KeyStone Architecture DDR3 Memory Controller User's Guide	SPRUGV8
KeyStone Architecture External Memory Interface (EMIF16) User's Guide	SPRUGZ3
Emulation and Trace Headers Technical Reference Manual	SPRU655
KeyStone Architecture Enhanced Direct Memory Access 3 (EDMA3) User's Guide	SPRUGS5
KeyStone Architecture General Purpose Input/Output (GPIO) User's Guide	SPRUGV1
Gigabit Ethernet (GbE) Switch Subsystem (1 GB) User's Guide	SPRUGV9
KeyStone Architecture Gigabit Ethernet (GbE) Switch Subsystem User's Guide	SPRUHJ5
KeyStone Architecture HyperLink User's Guide	SPRUGW8
Hardware Design Guide for KeyStone II Devices application report	SPRABV0
KeyStone Architecture Inter-IC control Bus (I^2C) User's Guide	SPRUGV3
KeyStone Architecture Memory Protection Unit (MPU) User's Guide	SPRUGW5
KeyStone Architecture Multicore Navigator User's Guide	SPRUGR9
KeyStone Architecture Multicore Shared Memory Controller (MSMC) User's Guide	SPRUGW7
KeyStone II Architecture Multicore Shared Memory Controller (MSMC) User's Guide	SPRUHJ6
KeyStone II Architecture Network Coprocessor (NETCP) for K2E and K2L Devices User's Guide	SPRUHZ0
Optimizing Application Software on KeyStone Devices application report	SPRABG8
KeyStone II Architecture Packet Accelerator 2 (PA2) for K2E and K2L Devices User's Guide	SPRUHZ2
KeyStone Architecture Peripheral Component Interconnect Express (PCIe) User's Guide	SPRUGS6
KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide	SPRUGV2
KeyStone Architecture Power Sleep Controller (PSC) User's Guide	SPRUGV4
KeyStone II Architecture Security Accelerator 2 (SA2) for K2E and K2L Devices User's Guide	SPRUHZ1
Security Addendum for KeyStone II Devices application report(1)	SPRABS4
KeyStone Architecture Semaphore2 Hardware Module User's Guide	SPRUGS3
KeyStone II Architecture Serializer/Deserializer (SerDes) User's Guide	SPRUHO3
KeyStone Architecture Serial Peripheral Interface (SPI) User's Guide	SPRUGP2
KeyStone Architecture Telecom Serial Interface Port (TSIP) User's Guide	SPRUGY4
KeyStone Architecture Universal Asynchronous Receiver/Transmitter (UART) User's Guide	SPRUGP1
KeyStone II Architecture Universal Serial Bus 3.0 (USB 3.0) User's Guide	SPRUHJ7
KeyStone II Architecture IQN2 User's Guide	SPRUH06

(1) Contact a TI sales office to obtain this document.

3.5 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to sample or buy.

Table 3-2. Related Links

PARTS	PRODUCT FOLDER	SAMPLE & BUY	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
AM5K2E04	Click here	Click here	Click here	Click here	Click here
AM5K2E02	Click here	Click here	Click here	Click here	Click here

3.6 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

TI E2E™ Online Community ***TI's Engineer-to-Engineer (E2E) Community.*** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

TI Embedded Processors Wiki ***Texas Instruments Embedded Processors Wiki.*** Established to help developers get started with Embedded Processors from Texas Instruments and to foster innovation and growth of general knowledge about the hardware and software surrounding these devices.

3.7 Trademarks

Code Composer Studio, XDS, E2E are trademarks of Texas Instruments.
MPCore is a trademark of ARM Ltd or its subsidiaries.
ARM, Cortex are registered trademarks of ARM Ltd or its subsidiaries.
All other trademarks are the property of their respective owners.

3.8 Electrostatic Discharge Caution

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

3.9 Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

4 ARM CorePac

The ARM CorePac is added in the AM5K2E0x to enable the ability for layer 2 and layer 3 processing on-chip. Operations such as traffic control, local O&M, NBAP/FP termination, and SCTP processing can all be performed with the Cortex-A15 processor core.

The ARM CorePac of the AM5K2E0x integrates one or more Cortex-A15 processor clusters with additional logic for bus protocol conversion, emulation, interrupt handling, and debug related enhancements. The Cortex-A15 processor is an ARMv7A-compatible, multi-issue out-of-order superscalar execution engine with integrated L1 caches. The implementation also supports advanced SIMDv2 (NEON technology) and VFPv4 (vector floating point) architecture extensions, security, virtualization, LPAE (large physical address extension), and multiprocessing extensions. The ARM CorePac includes an L2 cache and support for AMBA4 AXI and AXI coherence extension (ACE) protocols. An interrupt controller is included in the ARM CorePac to handle host interrupt requests in the system.

The ARM CorePac has three functional clock domains, including a high-frequency clock domain used by the Cortex-A15. The high-frequency domain is isolated from the rest of the device by asynchronous bridges.

The following figures show the ARM CorePac.

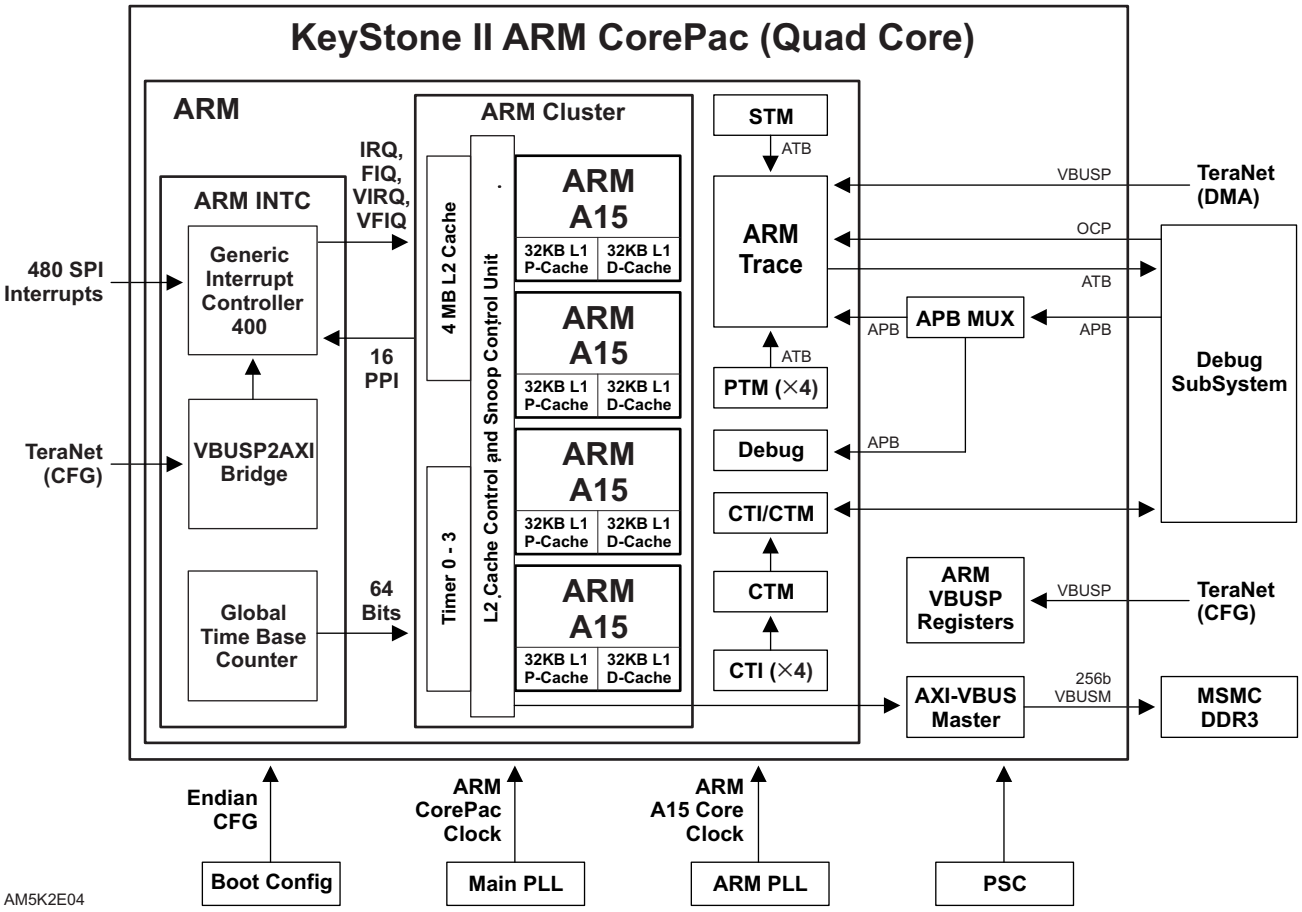


Figure 4-1. AM5K2E04 ARM CorePac Block Diagram

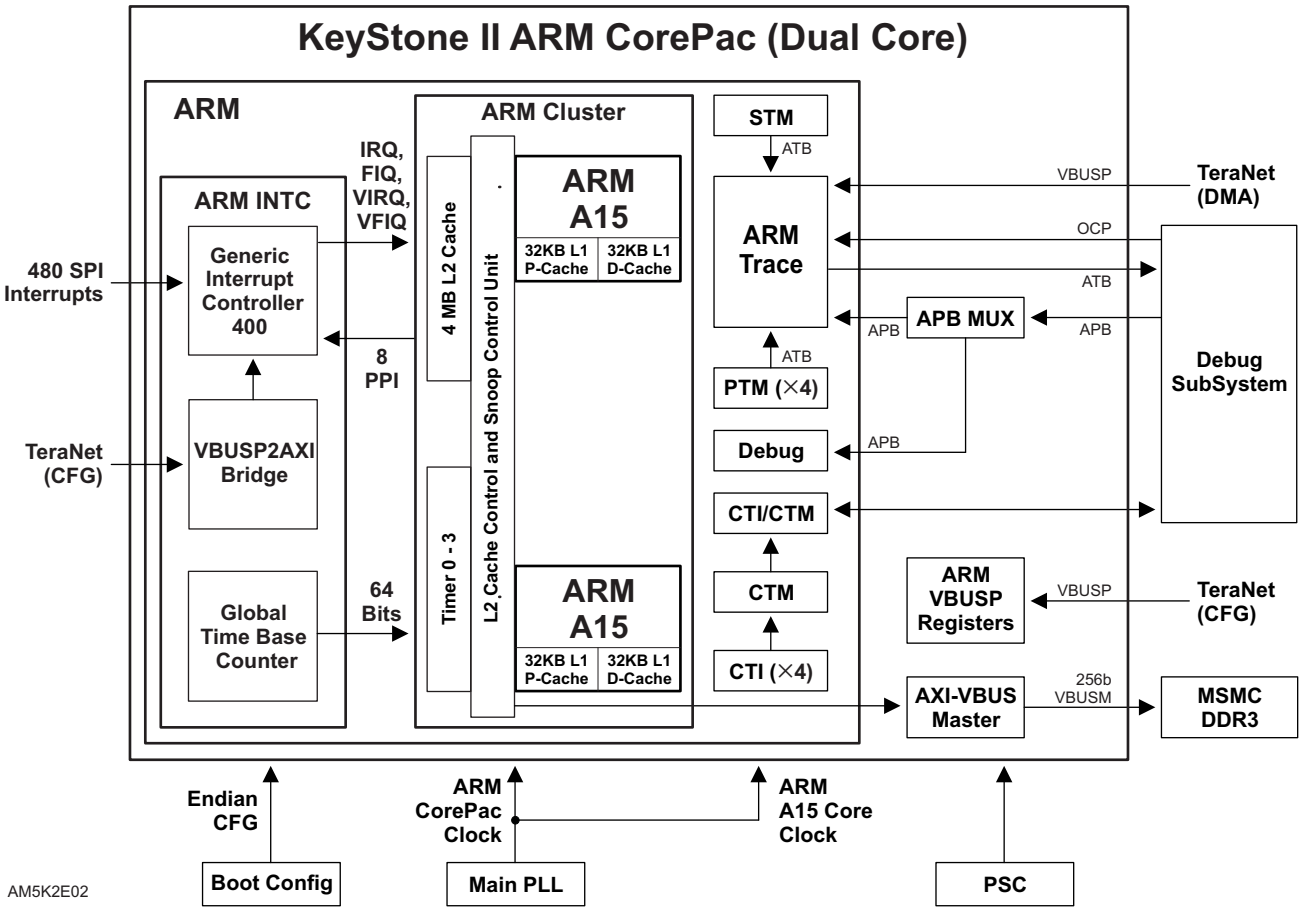


Figure 4-2. AM5K2E02 ARM CorePac Block Diagram

4.1 Features

The key features of the Quad Core ARM CorePac are as follows:

- One or more Cortex-A15 processors, each containing:
  - Cortex-A15 processor revision R2P4.
  - ARM architecture version 7 ISA.
  - Multi-issue, out-of-order, superscalar pipeline.
  - L1 and L2 instruction and data cache of 32KB, 2-way, 16 word line with 128-bit interface.
  - Integrated L2 cache of 4MB, 16-way, 16-word line, 128-bit interface to L1 along with ECC/parity.
  - Includes the NEON media coprocessor (NEON™), which implements the advanced SIMDv2 media processing architecture and the VFPv4 Vector Floating Point architecture.
  - The external interface uses the AXI protocol configured to 128-bit data width.
  - Includes the System Trace Macrocell (STM) support for non-invasive debugging.
  - Implements the ARMv7 debug with watchpoint and breakpoint registers and 32-bit advanced peripheral bus (APB) slave interface to CoreSight™ debug systems.
- Interrupt controller
  - Supports up to 480 interrupt requests
  - An integrated Global Time Base Counter (clocked by the SYSCLK divided by 6)
- Emulation/debug
  - Compatible with CoreSight™ architecture

4.2 System Integration

The ARM CorePac integrates the following group of submodules.

- **Cortex-A15 Processors:** Provides a high processing capability, including the NEON™ technology for mobile multimedia acceleration. The Cortex-A15 communicates with the rest of the ARM CorePac through an AXI bus with an AXI2VBUSM bridge and receives interrupts from the ARM CorePac interrupt controller (ARM INTC).
- **Interrupt Controller:** Handles interrupts from modules outside of the ARM CorePac (for details, see Section 4.3.3).
- **Clock Divider:** Provides the required divided clocks to the internal modules of the ARM CorePac and has a clock input from the Main PLL.
- **In-Circuit Emulator:** Fully compatible with CoreSight™ architecture and enables debugging capabilities.

4.3 ARM Cortex-A15 Processor

4.3.1 Overview

The ARM Cortex-A15 processor incorporates the technologies available in the ARM7™ architecture. These technologies include NEON™ for media and signal processing and Jazelle™ RCT for acceleration of real-time compilers, Thumb®-2 technology for code density, and the VFPv4 floating point architecture. For details, see the ARM Cortex-A15 Processor Technical Reference Manual.

4.3.2 Features

Table 4-1 shows the features supported by the Cortex-A15 processor core.

Table 4-1. Cortex-A15 Processor Core Supported Features

FEATURES	DESCRIPTION
ARM version 7-A ISA	Standard Cortex-A15 processor instruction set + Thumb2, ThumbEE, JazelleX Java accelerator, and media extensions
	Backward compatible with previous ARM ISA versions

Table 4-1. Cortex-A15 Processor Core Supported Features (continued)

FEATURES	DESCRIPTION
Cortex-A15 processor version	R2P4
Integer core	Main core for processing integer instructions
NEON core	Gives greatly enhanced throughput for media workloads and VFP-Lite support
Architecture Extensions	Security, virtualization and LPAE (40-bit physical address) extensions
L1 Lcache and Dcache	32KB, 2-way, 16 word line, 128 bit interface
L2 cache	4096KB, 16-way, 16 word line, 128 bit interface to L1, ECC/Parity is supported shared between cores
	L2 valid bits cleared by software loop or by hardware
Cache Coherency	Support for coherent memory accesses between A15 cores and other non-core master peripherals (Ex: EDMA) in the DDR3 and MSMC SRAM space.
Branch target address cache	Dynamic branch prediction with Branch Target Buffer (BTB) and Global History Buffer (GHB), a return stack, and an indirect predictor
Enhanced memory management unit	Mapping sizes are 4KB, 64KB, 1MB, and 16MB
Buses	128b AXI4 internal bus from Cortex-A15 converted to a 256b VBUSM to interface (through the MSMC) with MSMC SRAM, DDR EMIF, ROM, Interrupt controller and other system peripherals
Non-invasive Debug Support	Processor instruction trace using 4x Program Trace Macrocell (Coresight™ PTM), Data trace (print-f style debug) using System Trace Macrocell (Coresight™ STM) and Performance Monitoring Units (PMU)
Misc Debug Support	JTAG based debug and Cross triggering
Voltage	SmartReflex voltage domain for automatic voltage scaling
Power	Support for standby modes and separate core power domains for additional leakage power reduction

4.3.3 ARM Interrupt Controller

The ARM CorePac interrupt controller (AINTC) is responsible for prioritizing all service requests from the system peripherals and the secondary interrupt controller CIC2 and then generating either nIRQ or nFIQ to the Cortex-A15 processor. The type of the interrupt (nIRQ or nFIQ) and the priority of the interrupt inputs are programmable. The AINTC interfaces to the Cortex-A15 processor via the AXI port through an VBUS2AXI bridge and runs at half the processor speed. It has the capability to handle up to 480 requests, which can be steered/prioritized as A15 nFIQ or nIRQ interrupt requests.

The general features of the AINTC are:

- Up to 480 level sensitive shared peripheral interrupts (SPI) inputs
- Individual priority for each interrupt input
- Each interrupt can be steered to nFIQ or nIRQ
- Independent priority sorting for nFIQ and nIRQ
- Secure mask flag

On the chip level, there is a dedicated chip level interrupt controller to serve the ARM interrupt controller. See Section 6.3 for more details.

The figures below show an overall view of the ARM CorePac Interrupt Controller.

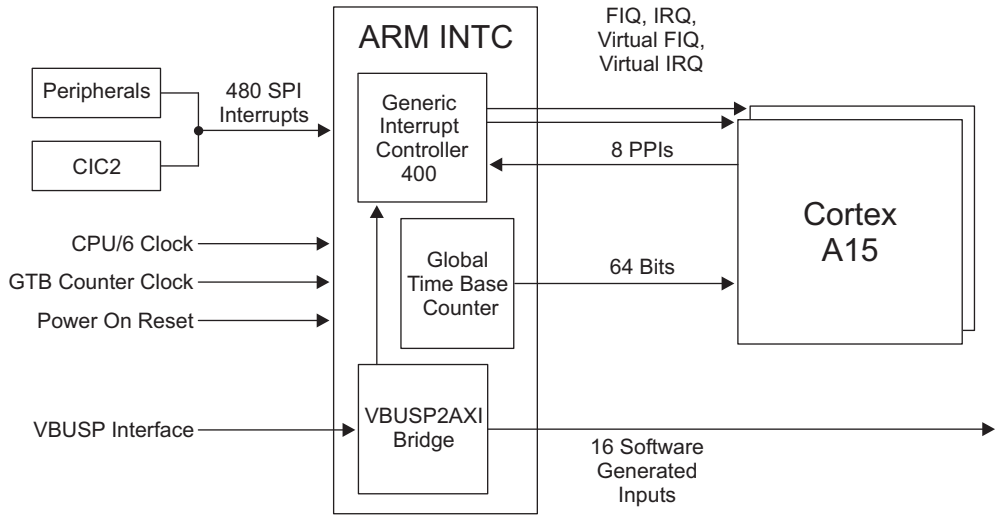


Figure 4-3. ARM Interrupt Controller for Two Cortex-A15 Processor Cores

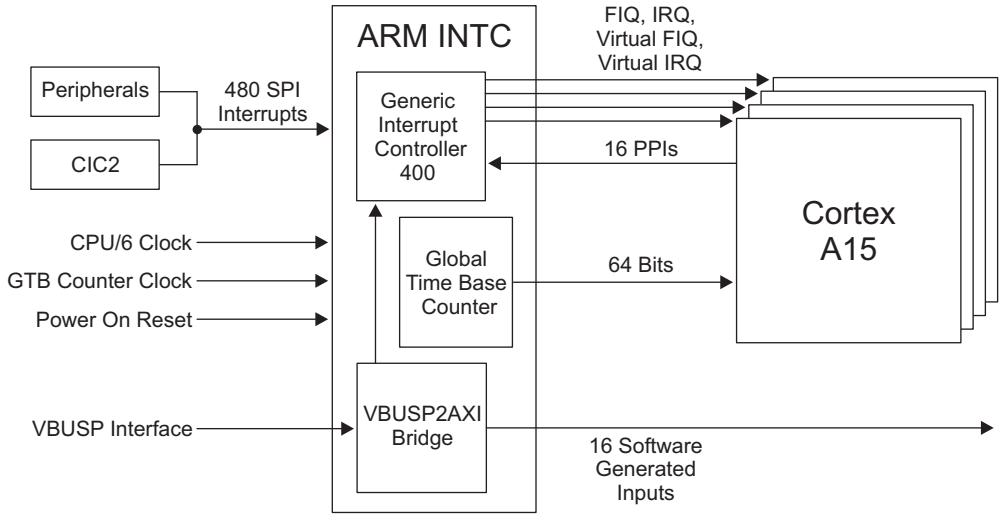


Figure 4-4. ARM Interrupt Controller for Four Cortex-A15 Processor Cores

4.3.4 Endianess

The ARM CorePac can operate in either little endian or big endian mode. When the ARM CorePac is in little endian mode and the rest of the system is in big endian mode, the bridges in the ARM CorePac are responsible for performing the endian conversion.

4.4 CFG Connection

The ARM CorePac has two slave ports. The AM5K2E0x masters cannot access the ARM CorePac internal memory space.

1. Slave port 0 (TeraNet 3P_A) is a 32 bit wide port used for the ARM Trace module.
2. Slave port 1 (TeraNet 3P_B) is a 32 bit wide port used to access the rest of the system configuration.

4.5 Main TeraNet Connection

There is one master port coming out of the ARM CorePac. The master port is a 256 bit wide port for the transactions going to the MSMC and DDR_EMIF data spaces.

4.6 Clocking and Reset

4.6.1 Clocking

The Cortex-A15 processor core clocks are sourced from the Controller. The Cortex-A15 processor core clock has a maximum frequency of 1.4 GHz. The ARM CorePac subsytem also uses the SYSCLK1 clock source from the main PLL which is locally divided (/1, /3 and /6) and provided to certain sub-modules inside the ARM CorePac. AINTC sub module runs at a frequency of SYSCLK1/6.

4.6.2 Reset

The ARM CorePac does not support local reset. It is reset whenever the device is under reset. In addition, the interrupt controller (AINTC) can only be reset during $\overline{\text{POR}}$ and $\overline{\text{RESETFULL}}$. AINTC also resets whenever device is under reset.

For the complete programming model, refer to the *KeyStone II Architecture ARM CorePac User's Guide* (SPRUHJ4).

5 Terminals

5.1 Package Terminals

Figure 5-1 shows the ABD 1089-ball grid array package (bottom view).

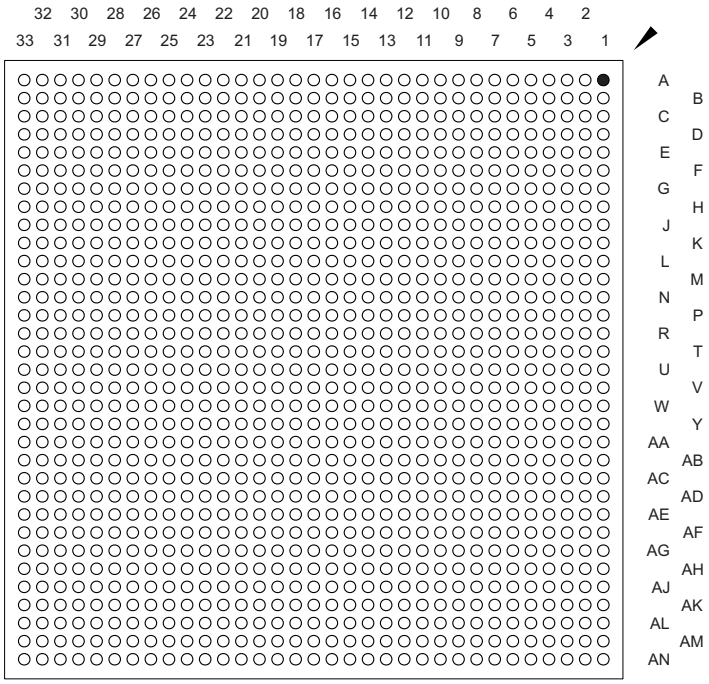


Figure 5-1. ABD 1089-Pin BGA Package (Bottom View)

5.2 Pin Map

The following figures show the AM5K2E0x pin assignments in four panels (A, B, C, and D).

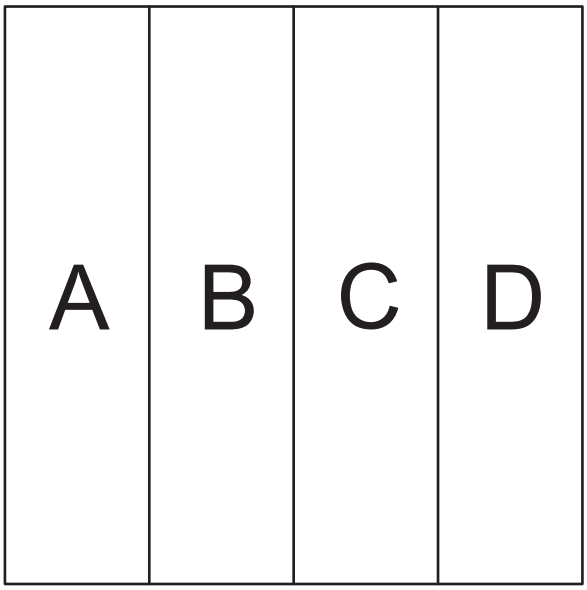


Figure 5-2. Pin Map Panels (Bottom View)

NOTE

XFI pins are associated with the 10-GbE feature and are supported only in the AM5K2E04 part.

	33	32	31	30	29	28	27	26
A	VSS	DVDD15	DDRDQM7	DDRDQS6P	DDRD48	DDRD49	DDRDQS5P	DDRD45
B	DVDD15	DDRD63	DDRD50	DDRDQS6N	DDRD52	DDRD44	DDRDQS5N	DDRD46
C	DDRDQS7P	DDRD61	DVDD15	DDRD53	VSS	DDRD42	VSS	DDRD47
D	DDRDQS7N	DDRD62	VSS	DDRD55	DVDD15	DDRD41	DVDD15	DDRDQM5
E	DDRD60	DDRD59	DDRDQM6	DDRD54	DDRD51	DDRD40	DDRD43	DDRDQM4
F	DDRD57	DDRD58	DDRD56	VSS	DVDD15	VSS	DVDD15	VSS
G	VCNTL1	VCNTL4	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15
H	USIMIO	RSV000	VCNTL2	VCNTL3	VCNTL0	VSS	DVDD15	VSS
J	USIMCLK	RSV001	VCNTL5	RSV013	RSV014	DVDD18	VSS	CVDD
K	TIMI0	TIMI1	TIMO0	TIMO1	USIMRST	VSS	DVDD18	VSSTMON
L	SPI2CLK	SPI0SCS3	SPI0SCS0	SPI2SCS3	SPI0SCS2	DVDD18	VSS	CVDDTMON
M	SPI1CLK	SPI1SCS1	SPI0DIN	SPI0CLK	SPI0SCS1	VSS	DVDD18	VSS
N	SPI1SCS3	VSS	DVDD18	SPI1SCS2	SPI0DOUT	DVDD18	VSS	CVDD
P	SPI2SCS2	SPI1DOUT	SPI2SCS0	SPI2DIN	SPI1SCS0	VSS	DVDD18	VSS
R	UART0CTS	UART0RTS	SPI2DOUT	SPI1DIN	SPI2SCS1	DVDD18	VSS	CVDD
T	UART1RTS	UART1TXD	UART1CTS	UART0TXD	UART0RXD	VSS	DVDD18	VSS
U	UART0DTR	VSS	DVDD18	VD	UART1RXD	DVDD18	VSS	CVDD
V	GPIO03	GPIO04	GPIO07	GPIO00	VCL	VSS	DVDD18	VSS
W	GPIO09	GPIO06	GPIO08	GPIO05	UART0DSR	DVDD18	VSS	CVDD
Y	GPIO14	GPIO12	GPIO15	GPIO11	GPIO02	VSS	DVDD18	VSS
AA	GPIO16	VSS	DVDD18	GPIO13	GPIO01	DVDD18	VSS	CVDD
AB	GPIO18	GPIO17	GPIO19	GPIO25	GPIO10	VSS	DVDD18	VSS
AC	GPIO21	GPIO24	GPIO23	GPIO26	GPIO20	DVDD18	VSS	CVDD
AD	GPIO28	GPIO27	GPIO29	GPIO31	GPIO22	VSS	DVDD18	VSS
AE	GPIO30	VSS	DVDD18	RSV030	$\overline{\text{RESET}}$	DVDD18	VSS	DVDD18
AF	RSV029	$\overline{\text{RESETFULL}}$	BOOTCOMPLETE	HOUT	TDO	VSS	DVDD18	VSS
AG	RSV028	$\overline{\text{TRST}}$	TDI	SCL0	SDA2	SDA0	VSS	DVDD18
AH	$\overline{\text{POR}}$	TMS	SCL2	SCL1	$\overline{\text{RESETSTAT}}$	VSS	DVDD18	VSS
AJ	TCK	SDA1	TSIP0FSA	DVDD18	VSS	RSV018	SGMII00REFRES	RSV019
AK	TSIP0CLKB	TSIP0FSB	TSIP0CLKA	VSS	SGMII0TXN1	SGMII0TXP1	VSS	SGMII0TXN3
AL	TSIP0TR1	TSIP0TX0	VSS	SGMII0TXN0	SGMII0TXP0	VSS	SGMII0TXN2	SGMII0TXP2
AM	DVDD18	TSIP0TX1	TSIP0TR0	VSS	SGMII0RXP1	SGMII0RXN1	VSS	SGMII0RXP3
AN	VSS	DVDD18	VSS	SGMII0RXP0	SGMII0RXN0	VSS	SGMII0RXP2	SGMII0RXN2
	33	**32**	**31**	**30**	**29**	**28**	**27**	**26**

Figure 5-3. AM5K2Ex Left End Panel (A) — Bottom View

	25	24	23	22	21	20	19	18
A	DDRD39	DDRDQS4P	DDRCB00	DDRDQS8N	DDRCB07	DDRCB06	DDRCKE1	DDRA08
B	DDRD38	DDRDQS4N	DDRD32	DDRDQS8P	DDRCB05	DDRDQM8	RSV021	DDRBA2
C	DVDD15	DDRD35	VSS	DDRCB04	DVDD15	DDRCKE0	VSS	DDRA14
D	VSS	DDRD34	DVDD15	DDRCB01	VSS	$\overline{\text{DDRRESET}}$	DVDD15	DDRA11
E	DDRD37	DDRD36	DDRD33	DDRCB02	DDRCB03	RSV022	DDRA15	DDRA12
F	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS
G	VSS	AVDDA10	VSS	AVDDA9	VSS	DVDD15	VSS	DDRRZQ2
H	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS
J	VSS	VNWA2	VSS	CVDD	VSS	CVDD	VSS	CVDD
K	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
L	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
M	CVDD	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS
N	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
P	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
R	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
T	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
U	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
V	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS
W	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
Y	CVDD	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS
AA	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS	CVDD
AB	CVDD	VSS	CVDD1	VSS	CVDD	VSS	CVDD	VSS
AC	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD
AD	CVDD	VSS	VNWA3	VSS	CVDD	VSS	CVDD	VSS
AE	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV
AF	VDDALV	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV	VSS
AG	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV
AH	VDDAHV	VSS	VDDAHV	VSS	VDDAHV	VSS	VDDAHV	VSS
AJ	SGMII0CLKN	SGMII0CLKP	VSS	SGMII01REFRES	VSS	RSV016	VSS	PCIE0CLKN
AK	SGMII0TXP3	VSS	SGMII0TXN5	SGMII0TXP5	VSS	SGMII0TXN7	SGMII0TXP7	VSS
AL	VSS	SGMII0TXN4	SGMII0TXP4	VSS	SGMII0TXN6	SGMII0TXP6	VSS	PCIE0TXN0
AM	SGMII0RXN3	VSS	SGMII0RXP5	SGMII0RXN5	VSS	SGMII0RXP7	SGMII0RXN7	VSS
AN	VSS	SGMII0RXP4	SGMII0RXN4	VSS	SGMII0RXP6	SGMII0RXN6	VSS	PCIE0RXP0
	25	24	23	22	21	20	19	18

Figure 5-4. AM5K2Ex Left Center Panel (B) — Bottom View

17	16	15	14	13	12	11	10	9	
DDRA06	DDRCLKOUTP1	DDRCLKOUTN1	RSV023	DDRA10	$\overline{\text{DDRCE0}}$	DDRD26	DDRDQS3P	DDRD31	**A**
DDRA09	DDRA02	DDRCLKOUTP0	DDRCLKOUTN0	$\overline{\text{DDRRAS}}$	$\overline{\text{DDRCAS}}$	DDRD25	DDRDQS3N	DDRD29	**B**
DVDD15	DDRA03	DDRA01	VSS	DDRBA1	$\overline{\text{DDRCE1}}$	DVDD15	DDRD27	VSS	**C**
VSS	DDRA04	DDRA00	DVDD15	DDRBA0	DDRODT0	VSS	DDRD28	DVDD15	**D**
DDRA07	DDRA05	DDRRZQ0	AVDDA7	$\overline{\text{DDRWE}}$	DDRA13	DDRODT1	DDRD24	DDRD30	**E**
DVDD15	VSS	DDRVREFSSTL	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	**F**
VSS	AVDDA8	VSS	DVDD15	VSS	DDRRZQ1	VSS	DVDD15	VSS	**G**
DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	VSS	DVDD15	**H**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDDCMON	VSSCMON	**J**
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VPP0	**K**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	**L**
CVDD	VSS	CVDD	VSS	CVDD1	VSS	CVDD1	VSS	VPP1	**M**
VSS	CVDD	VSS	CVDD	VSS	CVDD1	VSS	CVDD	VSS	**N**
CVDD	VSS	CVDD	VSS	CVDD	VSS	USB1DVDD33	VSS	VNWA1	**P**
VSS	CVDD	VSS	CVDD	VSS	VDDUSB1	VSS	USB1VPH	VSS	**R**
CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDUSB1	VSS	USB1VPTX	**T**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDUSB0	VSS	**U**
CVDD	VSS	CVDD	VSS	CVDD	VSS	VDDUSB0	VSS	USB0VPTX	**V**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	USB0VPH	VSS	**W**
CVDD	VSS	CVDD	VSS	CVDD1	VSS	USB0DVDD33	VSS	CVDD	**Y**
VSS	CVDD	VSS	CVDD1	VSS	CVDD1	VSS	CVDD	VSS	**AA**
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	**AB**
VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	**AC**
CVDD	VSS	CVDD	VSS	CVDD	VSS	CVDD	VSS	VNWA4	**AD**
VSS	VDDALV	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV	VSS	**AE**
VDDALV	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV	VSS	VDDALV	**AF**
VSS	VDDALV	VSS	PCIE0REFRES	VSS	RSV017	VSS	XFIREFRES0	VSS	**AG**
VDDAHV	VSS	VDDAHV	VSS	VDDAHV	VSS	VDDAHV	VSS	RSV020	**AH**
PCIE0CLKP	VSS	PCIE1CLKN	PCIE1CLKP	VSS	XFICLKN	XFICLKP	PCIE1REFRES	VSS	**AJ**
PCIE0TXN1	PCIE0TXP1	VSS	PCIE1TXN1	PCIE1TXP1	VSS	XFITXN1	XFITXP1	VSS	**AK**
PCIE0TXP0	VSS	PCIE1TXN0	PCIE1TXP0	VSS	XFITXN0	XFITXP0	VSS	HYPLNK0TXN0	**AL**
PCIE0RXP1	PCIE0RXN1	VSS	PCIE1RXP1	PCIE1RXN1	VSS	XFIRXP1	XFIRXN1	VSS	**AM**
PCIE0RXN0	VSS	PCIE1RXP0	PCIE1RXN0	VSS	XFIRXP0	XFIRXN0	VSS	HYPLNK0RXP0	**AN**
17	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	

Figure 5-5. AM5K2Ex Right Center Panel (C) — Bottom View

8	7	6	5	4	3	2	1	
DDRD22	DDRDQS2N	DDRD09	DDRD08	DDRDQS1P	DDRDQM0	DVDD15	VSS	**A**
DDRD23	DDRDQS2P	DDRD19	DDRD10	DDRDQS1N	DDRD15	DDRD07	DVDD15	**B**
DDRD21	DVDD15	DDRD20	VSS	DDRD12	DVDD15	DDRDQS0N	DDRDQS0P	**C**
DDRDQM2	VSS	DDRD16	DVDD15	DDRD13	VSS	DDRD05	DDRD01	**D**
DDRDQM3	DDRD18	DDRD17	DDRD11	DDRD14	DDRDQM1	DDRD00	DDRD03	**E**
VSS	DVDD15	VSS	DVDD15	VSS	DDRD06	DDRD02	DDRD04	**F**
AVDDA6	VSS	EMIFD09	EMIFD00	DDRCLKP	DDRCLKN	RSV004	RSV005	**G**
VSS	DVDD15	EMIFD06	EMIFD04	EMIFD13	EMIFD11	EMIFD08	EMIFD07	**H**
RSV010	VSS	EMIFD05	EMIFD15	EMIFD12	VSS	DVDD18	EMIFD03	**J**
RSV009	AVDDA2	EMIFD14	EMIFD10	EMIFD02	EMIFD01	EMIFA19	EMIFA20	**K**
DVDD18	VSS	EMIFA23	EMIFA17	EMIFA21	EMIFA14	EMIFA16	EMIFA11	**L**
VSS	DVDD18	EMIFA18	EMIFA09	EMIFA10	EMIFA06	DVDD18	VSS	**M**
DVDD18	VSS	EMIFA05	EMIFA08	EMIFA07	EMIFWAIT1	VSS	USB1RX0M	**N**
VSS	DVDD18	EMIFA04	EMIFA01	USB1DM	EMIFA00	USB1TX0P	USB1RX0P	**P**
USB1VP	VSS	$\overline{\text{EMIFBE1}}$	$\overline{\text{EMIFWE}}$	USB1DP	EMIFA22	USB1TX0M	VSS	**R**
VSS	DVDD18	EMIFWAIT0	EMIFA13	USBCLKM	EMIFA15	VSS	USB0RX0M	**T**
USB0VP	VSS	EMIFA02	EMIFA12	USBCLKP	VSS	USB0TX0M	USB0RX0P	**U**
VSS	DVDD18	EMIFA03	USB1RESREF	USB1VBUS	USB0DP	USB0TX0P	VSS	**V**
RSV011	RSV012	USB1ID0	USB0VBUS	USB0ID0	USB0DM	VSS	$\overline{\text{EMIFCE2}}$	**W**
VSS	DVDD18	USB0RESREF	EMU17	$\overline{\text{EMIFCE1}}$	EMIF$\overline{\text{RW}}$	$\overline{\text{EMIFOE}}$	$\overline{\text{EMIFCE3}}$	**Y**
DVDD18	VSS	EMU01	EMU15	VSS	DVDD18	$\overline{\text{EMIFBE0}}$	$\overline{\text{EMIFCE0}}$	**AA**
VSS	DVDD18	HYPLNK0TXPMCLK	EMU06	EMU08	EMU14	EMU16	USB0DRVVBUS	**AB**
DVDD18	VSS	HYPLNK0TXFLCLK	EMU00	EMU02	EMU07	EMU13	USB1DRVVBUS	**AC**
VSS	AVDDA3	EMU04	VSS	DVDD18	EMU05	EMU12	HYPLNK0RXPMDAT	**AD**
DVDD18	XFIMDIO	EMU03	TSSYNCEVT	SYSCLKOUT	HYPLNK0RXFLDAT	RSV002	HYPLNK0RXPMCLK	**AE**
VSS	AVDDA1	XFIMDCLK	HYPLNK0TXPMDAT	TSCOMPOUT	EMU10	RSV003	CORECLKP	**AF**
RSV015	VSS	NETCPCLKSEL	RSV008	EMU11	TSPUSHEVT0	TSRXCLKOUT1P	CORECLKN	**AG**
VSS	XFIREFRES1	MDCLK0	MDIO0	HYPLNK0TXFLDAT	EMU09	EMU18	TSRXCLKOUT1N	**AH**
HYPLNK0CLKN	HYPLNK0CLKP	VSS	HYPLNK0REFRES	VSS	HYPLNK0RXFLCLK	TSRXCLKOUT0N	TSRXCLKOUT0P	**AJ**
HYPLNK0TXN1	HYPLNK0TXP1	VSS	HYPLNK0TXN3	HYPLNK0TXP3	VSS	TSREFCLKP	TSREFCLKN	**AK**
HYPLNK0TXP0	VSS	HYPLNK0TXN2	HYPLNK0TXP2	VSS	RSV006	RSV007	TSPUSHEVT1	**AL**
HYPLNK0RXP1	HYPLNK0RXN1	VSS	HYPLNK0RXP3	HYPLNK0RXN3	VSS	NETCPCLKP	VSS	**AM**
HYPLNK0RXN0	VSS	HYPLNK0RXP2	HYPLNK0RXN2	VSS	NETCPCLKN	VSS	VSS	**AN**
8	**7**	**6**	**5**	**4**	**3**	**2**	**1**	

Figure 5-6. AM5K2Ex Right End Panel (D) — Bottom View

5.3 Terminal Functions

The terminal functions table (Table 5-2) identifies the external signal names, the associated pin (ball) numbers, the pin type (I, O/Z, or I/O/Z), whether the pin has any internal pullup/pulldown resistors, and gives functional pin descriptions. This table is arranged by function. The power terminal functions table (Table 5-3) lists the various power supply pins and ground pins and gives functional pin descriptions. Table 5-4 shows all pins arranged by signal name. Table 5-5 shows all pins arranged by ball number.

Some pins have additional functions beyond their primary functions. There are 21 pins that have a secondary function and 15 pins that have a tertiary function. Secondary functions are indicated with a superscript 2 (^{2}) and tertiary functions are indicated with a superscript 3 (^{3}).

For more detailed information on device configuration, peripheral selection, multiplexed/shared pins, and pullup/pulldown resistors, see Section 8.2.

Use the symbol definitions in Table 5-1 when reading Table 5-2.

Table 5-1. I/O Functional Symbol Definitions

FUNCTIONAL SYMBOL	DEFINITION	Table 5-2 COLUMN HEADING
IPD or IPU	Internal 100-µA pulldown or pullup is provided for this terminal. In most systems, a 1-kΩ resistor can be used to oppose the IPD/IPU.	IPD/IPU
A	Analog signal	Type
GND	Ground	Type
I	Input terminal	Type
O	Output terminal	Type
P	Power supply voltage	Type
Z	Three-state terminal or high impedance	Type

Table 5-2. Terminal Functions — Signals and Control by Function

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
Boot Configuration Pins				
BOOTMODE_RSVD[2]	Y31	I	Down	ARM Big Endian Configuration pin. Secondary function for GPIO15.
AVSIFSEL[0][2]	K33	I	Down	Default value (bootstrapped) for SR PINMUX Register (SR_PINCTL). Secondary function for TIMI0
AVSIFSEL[1][2]	K32	I	Down	Default value (bootstrapped) for SR PINMUX Register (SR_PINCTL). Secondary function for TIMI1
BOOTCOMPLETE	AF31	OZ	Down	Boot progress indication output
BOOTMODE00[2]	AA29	I	Down	User defined Boot Mode pin. Secondary function for GPIO01.
BOOTMODE01[2]	Y29	I	Down	User defined Boot Mode pin. Secondary function for GPIO02.
BOOTMODE02[2]	V33	I	Down	User defined Boot Mode pin. Secondary function for GPIO03.
BOOTMODE03[2]	V32	I	Down	User defined Boot Mode pin. Secondary function for GPIO04.
BOOTMODE04[2]	W30	I	Down	User defined Boot Mode pin. Secondary function for GPIO05.
BOOTMODE05[2]	W32	I	Down	User defined Boot Mode pin. Secondary function for GPIO06.
BOOTMODE06[2]	V31	I	Down	User defined Boot Mode pin. Secondary function for GPIO07.
BOOTMODE07[2]	W31	I	Down	User defined Boot Mode pin. Secondary function for GPIO08.
BOOTMODE08[2]	W33	I	Down	User defined Boot Mode pin. Secondary function for GPIO09.
BOOTMODE09[2]	AB29	I	Down	User defined Boot Mode pin. Secondary function for GPIO10.
BOOTMODE10[2]	Y30	I	Down	User defined Boot Mode pin. Secondary function for GPIO11.
BOOTMODE11[2]	Y32	I	Down	User defined Boot Mode pin. Secondary function for GPIO12.
BOOTMODE12[2]	AA30	I	Down	User defined Boot Mode pin. Secondary function for GPIO13.
BOOTMODE13[2]	AA33	I	Down	User defined Boot Mode pin. Secondary function for GPIO16.
BOOTMODE14[2]	AB32	I	Down	User defined Boot Mode pin. Secondary function for GPIO17.
BOOTMODE15[2]	AB33	I	Down	User defined Boot Mode pin. Secondary function for GPIO18.
LENDIAN[2]	V30	I	Up	Little Endian Configuration pin. Secondary function for GPIO00.

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
MAINPLLODSEL[2]	Y33	I	Down	Post divider select for main PLL.. Secondary function for GPIO14.
Clock / Reset				
CORECLKN	AG1	I		System clock input to main PLL
CORECLKP	AF1	I		
DDRCLKN	G3	I		DDR3 reference clock input to DDR PLL
DDRCLKP	G4	I		
HOUT	AF30	OZ	Up	Interrupt output pulse created by IPCGRH
HYPLNK0CLKN	AJ8	I		HyperLink reference clock to drive HyperLink SerDes
HYPLNK0CLKP	AJ7	I		
NETCPCLKN	AN3	I		NETCP sub-system reference clock
NETCPCLKP	AM2	I		
NETCPCLKSEL	AG6	I	Down	NETCP clock select to choose between core clock and NETCPCLK pins
PCIE0CLKN	AJ18	I		PCIe Clock input to drive PCIe0 SerDes
PCIE0CLKP	AJ17	I		
PCIE1CLKN	AJ15	I		PCIe Clock Input to drive PCIe1 SerDes
PCIE1CLKP	AJ14	I		
$\overline{\text{POR}}$	AH33	I		Power-on reset
$\overline{\text{RESETFULL}}$	AF32	I	Up	Full reset
$\overline{\text{RESETSTAT}}$	AH29	O	Up	Reset Status Output. Drives low during Power-on Reset (No HHV override). Available after core and IOs are completely powered-up.
$\overline{\text{RESET}}$	AE29	I	Up	Warm reset of non-isolated portion of the device
SGMII0CLKN	AJ25	I		SGMII reference clock to drive both SGMII0 SerDes SGMII reference clock to drive the SGMII SerDes
SGMII0CLKP	AJ24	I		
SYSCLKOUT	AE4	OZ	Down	System clock output to be used as a general purpose output clock for debug purposes
TSREFCLKN	AK1	I		Clock from external OCXO/VCXO for SyncE
TSREFCLKP	AK2	I		
TSRXCLKOUT0N	AJ2	O		SERDES recovered clock output for SyncE
TSRXCLKOUT0P	AJ1	O		
TSRXCLKOUT1N	AH1	O		SERDES recovered clock output for SyncE
TSRXCLKOUT1P	AG2	O		
USBCLKM	T4	I		USB0_3.0 reference clock
USBCLKP	U4	I		
XFICLKN	AJ12	I		XFI reference clock to drive the XFI SerDes
XFICLKP	AJ11	I		

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
DDR3				
DDRA00	D15	OZ		DDR3 EMIF address bus
DDRA01	C15	OZ		
DDRA02	B16	OZ		
DDRA03	C16	OZ		
DDRA04	D16	OZ		
DDRA05	E16	OZ		
DDRA06	A17	OZ		
DDRA07	E17	OZ		
DDRA08	A18	OZ		
DDRA09	B17	OZ		
DDRA10	A13	OZ		
DDRA11	D18	OZ		
DDRA12	E18	OZ		
DDRA13	E12	OZ		
DDRA14	C18	OZ		
DDRA15	E19	OZ		
DDRBA0	D13	OZ		DDR3 EMIF bank address
DDRBA1	C13	OZ		
DDRBA2	B18	OZ		
$\overline{\text{DDRCAS}}$	B12	OZ		DDR3 EMIF column address strobe
DDRCB00	A23	IOZ		DDR3 EMIF check bits
DDRCB01	D22	IOZ		
DDRCB02	E22	IOZ		
DDRCB03	E21	IOZ		
DDRCB04	C22	IOZ		
DDRCB05	B21	IOZ		
DDRCB06	A20	IOZ		
DDRCB07	A21	IOZ		
$\overline{\text{DDRCE0}}$	A12	OZ		DDR3 EMIF chip enable0
$\overline{\text{DDRCE1}}$	C12	OZ		DDR3 EMIF chip enable1
DDRCKE0	C20	OZ		DDR3 EMIF clock enable0
DDRCKE1	A19	OZ		DDR3 EMIF clock enable1
DDRCLKOUTN0	B14	OZ		DDR3 EMIF Output Clocks to drive SDRAMs for Rank0
DDRCLKOUTP0	B15	OZ		
DDRCLKOUTN1	A15	OZ		DDR3 EMIF Output Clocks to drive SDRAMs for Rank1
DDRCLKOUTP1	A16	OZ		

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
DDRD00	E2	IOZ		DDR3 EMIF data bus
DDRD01	D1	IOZ		
DDRD02	F2	IOZ		
DDRD03	E1	IOZ		
DDRD04	F1	IOZ		
DDRD05	D2	IOZ		
DDRD06	F3	IOZ		
DDRD07	B2	IOZ		
DDRD08	A5	IOZ		
DDRD09	A6	IOZ		
DDRD10	B5	IOZ		
DDRD11	E5	IOZ		
DDRD12	C4	IOZ		
DDRD13	D4	IOZ		
DDRD14	E4	IOZ		
DDRD15	B3	IOZ		
DDRD16	D6	IOZ		DDR3 EMIF data bus
DDRD17	E6	IOZ		
DDRD18	E7	IOZ		
DDRD19	B6	IOZ		
DDRD20	C6	IOZ		
DDRD21	C8	IOZ		
DDRD22	A8	IOZ		
DDRD23	B8	IOZ		
DDRD24	E10	IOZ		
DDRD25	B11	IOZ		
DDRD26	A11	IOZ		
DDRD27	C10	IOZ		
DDRD28	D10	IOZ		
DDRD29	B9	IOZ		
DDRD30	E9	IOZ		
DDRD31	A9	IOZ		
DDRD32	B23	IOZ		DDR3 EMIF data bus
DDRD33	E23	IOZ		
DDRD34	D24	IOZ		
DDRD35	C24	IOZ		
DDRD36	E24	IOZ		
DDRD37	E25	IOZ		
DDRD38	B25	IOZ		
DDRD39	A25	IOZ		
DDRD40	E28	IOZ		
DDRD41	D28	IOZ		
DDRD42	C28	IOZ		
DDRD43	E27	IOZ		
DDRD44	B28	IOZ		
DDRD45	A26	IOZ		
DDRD46	B26	IOZ		
DDRD47	C26	IOZ		

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
DDRD48	A29	IOZ		DDR3 EMIF data bus
DDRD49	A28	IOZ		
DDRD50	B31	IOZ		
DDRD51	E29	IOZ		
DDRD52	B29	IOZ		
DDRD53	C30	IOZ		
DDRD54	E30	IOZ		
DDRD55	D30	IOZ		
DDRD56	F31	IOZ		
DDRD57	F33	IOZ		
DDRD58	F32	IOZ		
DDRD59	E32	IOZ		
DDRD60	E33	IOZ		
DDRD61	C32	IOZ		
DDRD62	D32	IOZ		
DDRD63	B32	IOZ		
DDRDQM0	A3	OZ		DDR3 EMIF Data Masks
DDRDQM1	E3	OZ		
DDRDQM2	D8	OZ		
DDRDQM3	E8	OZ		
DDRDQM4	E26	OZ		
DDRDQM5	D26	OZ		
DDRDQM6	E31	OZ		
DDRDQM7	A31	OZ		
DDRDQM8	B20	OZ		
DDRDQS0N	C2	IOZ	Up/Dn	DDR3 EMIF data strobe. Programmable pull-up/dn 350-650 ohm.
DDRDQS0P	C1	IOZ	Up/Dn	
DDRDQS1N	B4	IOZ	Up/Dn	
DDRDQS1P	A4	IOZ	Up/Dn	
DDRDQS2N	A7	IOZ	Up/Dn	
DDRDQS2P	B7	IOZ	Up/Dn	
DDRDQS3N	B10	IOZ	Up/Dn	
DDRDQS3P	A10	IOZ	Up/Dn	
DDRDQS4N	B24	IOZ	Up/Dn	
DDRDQS4P	A24	IOZ	Up/Dn	
DDRDQS5N	B27	IOZ	Up/Dn	
DDRDQS5P	A27	IOZ	Up/Dn	
DDRDQS6N	B30	IOZ	Up/Dn	
DDRDQS6P	A30	IOZ	Up/Dn	
DDRDQS7N	D33	IOZ	Up/Dn	
DDRDQS7P	C33	IOZ	Up/Dn	
DDRDQS8N	A22	IOZ	Up/Dn	
DDRDQS8P	B22	IOZ	Up/Dn	
DDRODT0	D12	OZ		DDR3 EMIF on-die termination outputs used to set termination on the SDRAMs
DDRODT1	E11	OZ		DDR3 EMIF on-die termination outputs used to set termination on the SDRAMs
$\overline{\text{DDRRAS}}$	B13	OZ		DDR3 EMIF row address strobe
$\overline{\text{DDRRESET}}$	D20	OZ		DDR3 reset signal. IO will work in LVCMOS mode to comply with JEDEC standard.
DDRRZQ0	E15	A		PTV compensation reference resistor pin for DDR3
DDRRZQ1	G12	A		PTV compensation reference resistor pin for DDR3
DDRRZQ2	G18	A		PTV compensation reference resistor pin for DDR3
$\overline{\text{DDRWE}}$	E13	OZ		DDR3 EMIF write enable

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
EMIF				
EMIFA00	P3	O	Down	EMIF address
EMIFA01	P5	O	Down	
EMIFA02	U6	O	Down	
EMIFA03	V6	O	Down	
EMIFA04	P6	O	Down	
EMIFA05	N6	O	Down	
EMIFA06	M3	O	Down	
EMIFA07	N4	O	Down	
EMIFA08	N5	O	Down	
EMIFA09	M5	O	Down	
EMIFA10	M4	O	Down	
EMIFA11	L1	O	Down	
EMIFA12	U5	O	Down	EMIF address
EMIFA13	T5	O	Down	
EMIFA14	L3	O	Down	
EMIFA15	T3	O	Down	
EMIFA16	L2	O	Down	
EMIFA17	L5	O	Down	
EMIFA18	M6	O	Down	
EMIFA19	K2	O	Down	
EMIFA20	K1	O	Down	
EMIFA21	L4	O	Down	
EMIFA22	R3	O	Down	
EMIFA23	L6	O	Down	
$\overline{\text{EMIFBE0}}$	AA2	O	Up	EMIF control signals
$\overline{\text{EMIFBE1}}$	R6	O	Up	
$\overline{\text{EMIFCE0}}$	AA1	O	Up	
$\overline{\text{EMIFCE1}}$	Y4	O	Up	
$\overline{\text{EMIFCE2}}$	W1	O	Up	
$\overline{\text{EMIFCE3}}$	Y1	O	Up	
$\overline{\text{EMIFOE}}$	Y2	O	Up	
EMIF$\overline{\text{RW}}$	Y3	O	Up	
EMIFWAIT0	T6	I	Down	
EMIFWAIT1	N3	I	Down	
$\overline{\text{EMIFWE}}$	R5	O	Up	

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
EMIFD00	G5	IOZ	Down	EMIF data
EMIFD01	K3	IOZ	Down	
EMIFD02	K4	IOZ	Down	
EMIFD03	J1	IOZ	Down	
EMIFD04	H5	IOZ	Down	
EMIFD05	J6	IOZ	Down	
EMIFD06	H6	IOZ	Down	
EMIFD07	H1	IOZ	Down	
EMIFD08	H2	IOZ	Down	
EMIFD09	G6	IOZ	Down	
EMIFD10	K5	IOZ	Down	
EMIFD11	H3	IOZ	Down	
EMIFD12	J4	IOZ	Down	
EMIFD13	H4	IOZ	Down	
EMIFD14	K6	IOZ	Down	
EMIFD15	J5	IOZ	Down	
EMU				
EMU00	AC5	IOZ	Up	Emulation and trace port
EMU01	AA6	IOZ	Up	
EMU02	AC4	IOZ	Up	
EMU03	AE6	IOZ	Up	
EMU04	AD6	IOZ	Up	
EMU05	AD3	IOZ	Up	
EMU06	AB5	IOZ	Up	
EMU07	AC3	IOZ	Up	
EMU08	AB4	IOZ	Up	
EMU09	AH3	IOZ	Up	
EMU10	AF3	IOZ	Up	
EMU11	AG4	IOZ	Up	
EMU12	AD2	IOZ	Up	
EMU13	AC2	IOZ	Up	
EMU14	AB3	IOZ	Up	
EMU15	AA5	IOZ	Up	
EMU16	AB2	IOZ	Up	
EMU17	Y5	IOZ	Up	
EMU18	AH2	IOZ	Up	
EMU19[3]	AB32	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO17.
EMU20[3]	AB33	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO18.
EMU21[3]	AB31	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO19.
EMU22[3]	AC29	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO20.
EMU23[3]	AC33	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO21.
EMU24[3]	AD29	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO22.
EMU25[3]	AC31	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO23.
EMU26[3]	AC32	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO24.
EMU27[3]	AB30	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO25.
EMU28[3]	AC30	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO26.
EMU29[3]	AD32	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO27.
EMU30[3]	AD33	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO28.
EMU31[3]	AD31	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO29.
EMU32[3]	AE33	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO30.

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
EMU33[3]	AD30	IOZ	Down	EMU (Unique select for EMU muxing on each GPIO pin.) Tertiary function for GPIO31.
General Purpose Input/Output (GPIO)				
GPIO00	V30	IOZ	Up	GPIOs
GPIO01	AA29	IOZ	Down	
GPIO02	Y29	IOZ	Down	
GPIO03	V33	IOZ	Down	
GPIO04	V32	IOZ	Down	
GPIO05	W30	IOZ	Down	
GPIO06	W32	IOZ	Down	
GPIO07	V31	IOZ	Down	
GPIO08	W31	IOZ	Down	
GPIO09	W33	IOZ	Down	
GPIO10	AB29	IOZ	Down	
GPIO11	Y30	IOZ	Down	
GPIO12	Y32	IOZ	Down	
GPIO13	AA30	IOZ	Down	
GPIO14	Y33	IOZ	Down	
GPIO15	Y31	IOZ	Down	
GPIO16	AA33	IOZ	Down	GPIOs
GPIO17	AB32	IOZ	Down	
GPIO18	AB33	IOZ	Down	
GPIO19	AB31	IOZ	Down	
GPIO20	AC29	IOZ	Down	
GPIO21	AC33	IOZ	Down	
GPIO22	AD29	IOZ	Down	
GPIO23	AC31	IOZ	Down	
GPIO24	AC32	IOZ	Down	
GPIO25	AB30	IOZ	Down	
GPIO26	AC30	IOZ	Down	
GPIO27	AD32	IOZ	Down	
GPIO28	AD33	IOZ	Down	
GPIO29	AD31	IOZ	Down	
GPIO30	AE33	IOZ	Down	
GPIO31	AD30	IOZ	Down	
HyperLink				
HYPLNK0RXN0	AN8	I		HyperLink receive data
HYPLNK0RXN1	AM7	I		
HYPLNK0RXN2	AN5	I		
HYPLNK0RXN3	AM4	I		
HYPLNK0RXP0	AN9	I		
HYPLNK0RXP1	AM8	I		
HYPLNK0RXP2	AN6	I		
HYPLNK0RXP3	AM5	I		
HYPLNK0TXN0	AL9	O		HyperLink transmit data
HYPLNK0TXN1	AK8	O		
HYPLNK0TXN2	AL6	O		
HYPLNK0TXN3	AK5	O		
HYPLNK0TXP0	AL8	O		
HYPLNK0TXP1	AK7	O		
HYPLNK0TXP2	AL5	O		
HYPLNK0TXP3	AK4	O		

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
HYPLNK0RXFLCLK	AJ3	O	down	HyperLink sideband signals
HYPLNK0RXFLDAT	AE3	O	down	
HYPLNK0RXPMCLK	AE1	I	down	
HYPLNK0RXPMDAT	AD1	I	down	
HYPLNK0TXFLCLK	AC6	I	down	
HYPLNK0TXFLDAT	AH4	I	down	
HYPLNK0TXPMCLK	AB6	O	down	
HYPLNK0TXPMDAT	AF5	O	down	
HYPLNK0REFRES	AJ5	A		HyperLink SerDes reference resistor input (3 kΩ ±1%)
I²C				
SCL0	AG30	IOZ		I²C0 clock
SCL1	AH30	IOZ		I²C1 clock
SCL2	AH31	IOZ		I²C2 clock
SDA0	AG28	IOZ		I²C0 data
SDA1	AJ32	IOZ		I²C1 data
SDA2	AG29	IOZ		I²C2 data
JTAG				
TCK	AJ33	I	Up	JTAG clock input
TDI	AG31	I	Up	JTAG data input
TDO	AF29	OZ	Up	JTAG data output
TMS	AH32	I	Up	JTAG test mode input
$\overline{\text{TRST}}$	AG32	I	Down	JTAG reset
MDIO				
MDCLK0	AH6	O	Down	MDIO0 Clock
MDIO0	AH5	IOZ	Up	MDIO0 Data
XFIMDCLK	AF6	O	Down	XFI MDIO Clock
XFIMDIO	AE7	IOZ	Up	XFI MDIO Data
PCIe				
PCIE0REFRES	AG14	A		PCIexpress0 SerDes reference resistor input (3 kΩ ±1%)
PCIE0RXN0	AN17	I		PCIexpress0 lane 0 receive data
PCIE0RXP0	AN18	I		
PCIE0RXN1	AM16	I		PCIexpress0 lane 1 receive data
PCIE0RXP1	AM17	I		
PCIE0TXN0	AL18	O		PCIexpress0 lane 0 transmit data
PCIE0TXP0	AL17	O		
PCIE0TXN1	AK17	O		PCIexpress0 lane 1 transmit data
PCIE0TXP1	AK16	O		
PCIE1REFRES	AJ10	A		PCIexpress1 SerDes reference resistor input (3 kΩ ±1%)
PCIE1RXN0	AN14	I		PCIexpress1lane 0 receive data
PCIE1RXP0	AN15	I		
PCIE1RXN1	AM13	I		PCIexpress1lane 1 receive data
PCIE1RXP1	AM14	I		
PCIE1TXN0	AL15	O		PCIexpress1 lane 0 transmit data
PCIE1TXP0	AL14	O		
PCIE1TXN1	AK14	O		PCIexpress1 lane 1 transmit data
PCIE1TXP1	AK13	O		
SGMII				
SGMII00REFRES	AJ27	A		SGMII0 SerDes reference resistor input (3 kΩ ±1%)
SGMII01REFRES	AJ22	A		SGMII1 SerDes reference resistor input (3 kΩ ±1%)
SGMII0RXN0	AN29	I		Ethernet MAC SGMII0 port 0 receive data
SGMII0RXP0	AN30	I		

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
SGMII0RXN1	AM28	I		Ethernet MAC SGMII0 port 1 receive data
SGMII0RXP1	AM29	I		
SGMII0RXN2	AN26	I		Ethernet MAC SGMII0 port 2 receive data
SGMII0RXP2	AN27	I		
SGMII0RXN3	AM25	I		Ethernet MAC SGMII0 port 3 receive data
SGMII0RXP3	AM26	I		
SGMII0RXN4	AN23	I		Ethernet MAC SGMII1 port 4 receive data
SGMII0RXP4	AN24	I		
SGMII0RXN5	AM22	I		Ethernet MAC SGMII1 port 5 receive data
SGMII0RXP5	AM23	I		
SGMII0RXN6	AN20	I		Ethernet MAC SGMII1 port 6 receive data
SGMII0RXP6	AN21	I		
SGMII0RXN7	AM19	I		Ethernet MAC SGMII1 port 7 receive data
SGMII0RXP7	AM20	I		
SGMII0TXN0	AL30	O		Ethernet MAC SGMII0 port 0 transmit data
SGMII0TXP0	AL29	O		
SGMII0TXN1	AK29	O		Ethernet MAC SGMII0 port 1 transmit data
SGMII0TXP1	AK28	O		
SGMII0TXN2	AL27	O		Ethernet MAC SGMII0 port 2 transmit data
SGMII0TXP2	AL26	O		
SGMII0TXN3	AK26	O		Ethernet MAC SGMII0 port 3 transmit data
SGMII0TXP3	AK25	O		
SGMII0TXN4	AL24	O		Ethernet MAC SGMII1 port 4 transmit data
SGMII0TXP4	AL23	O		
SGMII0TXN5	AK23	O		Ethernet MAC SGMII1 port 5 transmit data
SGMII0TXP5	AK22	O		
SGMII0TXN6	AL21	O		Ethernet MAC SGMII1 port 6 transmit data
SGMII0TXP6	AL20	O		
SGMII0TXN7	AK20	O		Ethernet MAC SGMII1 port 7 transmit data
SGMII0TXP7	AK19	O		
SmartReflex				
VCL	V29	IOZ		Voltage control I^2C clock
VCNTL0	H29	OZ		Voltage control outputs to variable core power supply
VCNTL1	G33	OZ		
VCNTL2	H31	OZ		
VCNTL3	H30	OZ		
VCNTL4	G32	OZ		
VCNTL5	J31	OZ		
VD	U30	IOZ		Voltage control I^2C data
SPI0				
SPI0CLK	M30	OZ	Down	SPI0 clock
SPI0DIN	M31	I	Down	SPI0 data in
SPI0DOUT	N29	OZ	Down	SPI0 data out
SPI0SCS0	L31	OZ	Up	SPI0 interface enable 0
SPI0SCS1	M29	OZ	Up	SPI0 interface enable 1
SPI0SCS2	L29	OZ	Up	SPI0 interface enable 2
SPI0SCS3	L32	OZ	Up	SPI0 interface enable 3
SPI1				
SPI1CLK	M33	OZ	Down	SPI1 clock
SPI1DIN	R30	I	Down	SPI1 data in
SPI1DOUT	P32	OZ	Down	SPI1 data out

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
SPI1SCS0	P29	OZ	Up	SPI1 interface enable 0
SPI1SCS1	M32	OZ	Up	SPI1 interface enable 1
SPI1SCS2	N30	OZ	Up	SPI1 interface enable 2
SPI1SCS3	N33	OZ	Up	SPI1 interface enable 3
SPI2				
SPI2CLK	L33	OZ	Down	SPI2 clock
SPI2DIN	P30	I	Down	SPI2 data in
SPI2DOUT	R31	OZ	Down	SPI2 data out
SPI2SCS0	P31	OZ	Up	SPI2 interface enable 0
SPI2SCS1	R29	OZ	Up	SPI2 interface enable 1
SPI2SCS2	P33	OZ	Up	SPI2 interface enable 2
SPI2SCS3	L30	OZ	Up	SPI2 interface enable 3
Sync-Ethernet / IEEE1588				
TSCOMPOUT	AF4	O	Down	IEEE1588 compare output
TSPUSHEVT0	AG3	I	Down	PPS push event from GPS for IEEE1588
TSPUSHEVT1	AL1	I	Down	Push event from BCN for IEEE1588
TSSYNCEVT	AE5	O	Down	IEEE1588 sync event output
Timer				
TIMI0	K33	I	Down	Timer 0 input
TIMO0	K31	OZ	Down	Timer 0 output
TIMI1	K32	I	Down	Timer 1 input
TIMO1	K30	OZ	Down	Timer 1 output
TSIP				
TSIP0CLKA	AK31	I	Down	CLKA0 TSIP0 external clock A
TSIP0CLKB	AK33	I	Down	CLKB0 TSIP0 external clock B
TSIP0FSA	AJ31	I	Down	FSA0 TSIP0 frame sync A
TSIP0FSB	AK32	I	Down	FSB0 TSIP0 frame sync B
TSIP0TR0	AM31	I	Down	TR00 TR01 TSIP0 receive data
TSIP0TR1	AL33	I	Down	
TSIP0TX0	AL32	OZ	Down	TX00 TX01 TSIP0 transmit data
TSIP0TX1	AM32	OZ	Down	
UART0				
UART0CTS	R33	I	Down	UART0 clear to send
UART0DSR	W29	I	Down	UART0 data set ready
UART0DTR	U33	OZ	Down	UART0 data terminal ready
UART0RTS	R32	OZ	Down	UART0 request to send
UART0RXD	T29	I	Down	UART0 serial data in
UART0TXD	T30	OZ	Down	UART0 serial data out
UART1				
UART1CTS	T31	I	Down	UART1 clear to send
UART1RTS	T33	OZ	Down	UART1 request to send
UART1RXD	U29	I	Down	UART1 serial data in
UART1TXD	T32	OZ	Down	UART1 serial data out
USB0 (USB_3.0)				
USB0DM	W3	IOZ		USB0 D-
USB0DP	V3	IOZ		USB0 D+
USB0DRVVBUS	AB1	O	Down	USB0 DRVVBUS output
USB0ID0	W4	A		USB0 ID
USB0RESREF	Y6	A		Reference resistor connection for USB0 PHY (200 Ω +- 1% resistor to ground)
USB0RX0M	T1	I		USB0_3.0 receive data
USB0RX0P	U1	I		

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
USB0TX0M	U2	O		USB0_3.0 transmit data
USB0TX0P	V2	O		
USB0VBUS	W5	A		USB0 5-V analog input. Connect to VBUS pin on USB connector through protection switch
USB1 (USB_3.0)				
USB1DM	P4	IOZ		USB1 D-
USB1DP	R4	IOZ		USB1 D+
USB1DRVVBUS	AC1	O	Down	USB1 DRVVBUS output
USB1ID0	W6	A		USB1 ID
USB1RESREF	V5	A		Reference resistor connection for USB1 PHY (200Ω +- 1% resistor to ground)
USB1RX0M	N1	I		USB1_3.0 receive data
USB1RX0P	P1	I		
USB1TX0M	R2	O		USB1_3.0 transmit data
USB1TX0P	P2	O		
USB1VBUS	V4	A		USB1 5-V analog input. Connect to VBUS pin on USB connector through protection switch
USIM				
USIMCLK	J33	OZ	Down	USIM clock
USIMIO	H33	IOZ	Up	USIM data
USIMRST	K29	OZ	Down	USIM reset
XFI (AM5K2E04 only)				
XFIRXN0	AN11	I		Ethernet MAC XFI port 0 receive data
XFIRXP0	AN12	I		
XFIRXN1	AM10	I		Ethernet MAC XFI port 1 receive data
XFIRXP1	AM11	I		
XFITXN0	AL12	O		Ethernet MAC XFI port 0 transmit data
XFITXP0	AL11	O		
XFITXN1	AK11	O		Ethernet MAC XFI port 1 transmit data
XFITXP1	AK10	O		
XFIREFRES0	AG10	A		XFI port 0 SerDes reference resistor input (3 kΩ ±1%)
XFIREFRES1	AH7	A		XFI port 1 SerDes reference resistor input (3 kΩ ±1%)

Table 5-2. Terminal Functions — Signals and Control by Function (continued)

SIGNAL NAME	BALL NO.	TYPE	IPD/IPU	DESCRIPTION
Reserved				
RSV000	H32	OZ	Down	Leave unconnected
RSV001	J32	OZ	Down	Leave unconnected
RSV002	AE2	O		Leave unconnected
RSV003	AF2	O		Leave unconnected
RSV004	G2	O		Leave unconnected
RSV005	G1	O		Leave unconnected
RSV006	AL3	O		Leave unconnected
RSV007	AL2	O		Leave unconnected
RSV008	AG5	OZ	Down	Leave unconnected
RSV009	K8	A		Connect to GND
RSV010	J8	A		Leave unconnected
RSV011	W8	A		Leave unconnected
RSV012	W7	A		Leave unconnected
RSV013	J30	A		Leave unconnected
RSV014	J29	A		Leave unconnected
RSV015	AG8	A		Leave unconnected
RSV016	AJ20	A		Leave unconnected
RSV017	AG12	A		Leave unconnected
RSV018	AJ28	A		Leave unconnected
RSV019	AJ26	A		Leave unconnected
RSV020	AH9	A		Leave unconnected
RSV021	B19	OZ		Leave unconnected
RSV022	E20	OZ		Leave unconnected
RSV023	A14	A		Leave unconnected
RSV028	AG33	I		Leave unconnected
RSV029	AF33	I		Leave unconnected
RSV030	AE30	I		Leave unconnected

Table 5-3. Terminal Functions — Power and Ground

SUPPLY	BALL NO.	VOLTS	DESCRIPTION
AVDDA1	AF7	1.8 V	COREPLL supply
AVDDA2	K7	1.8 V	NETCPPLL supply
AVDDA3	AD7	1.8 V	DDRPLL supply
AVDDA6	G8	1.8 V	DDRA DLL supply
AVDDA7	E14	1.8 V	DDRA DLL supply
AVDDA8	G16	1.8 V	DDRA DLL supply
AVDDA9	G22	1.8 V	DDRA DLL supply
AVDDA10	G24	1.8 V	DDRA DLL supply
CVDD	J12, J14, J16, J18, J20, J22, J26, K11, K13, K15, K17, K19, K21, K23, K25, L10, L12, L14, L16, L18, L20, L22, M15, M17, M19, M21, M25, N10, N14, N16, N18, N20, N22, N26, P13, P15, P17, P19, P21, P23, P25, R14, R16, R18, R20, R22, R24, R26, T13, T15, T17, T19, T21, T23, T25, U12, U14, U16, U18, U20, U22, U24, U26, V13, V15, V17, V19, V21, V23, V25, W12, W14, W16, W18, W20, W22, W24, W26, Y9, Y15, Y17, Y19, Y21, Y25, AA10, AA16, AA18, AA20, AA22, AA26, AB9, AB11, AB13, AB15, AB17, AB19, AB21, AB25, AC10, AC12, AC14, AC16, AC18, AC20, AC22, AC24, AC26, AD11, AD13, AD15, AD17, AD19, AD21, AD25	AVS	Smart Reflex core supply voltage
CVDD1	L24, M11, M13, M23, N12, N24, Y13, Y23, AA12, AA14, AA24, AB23	0.95 V	Core supply voltage for memory array
CVDDCMON	J10	AVS	CVDD Supply Monitor
CVDDTMON	L26	AVS	CVDD Supply Monitor
DDR3VREFSSTL	F15	DVDD15/2	DDR3 reference voltage
DVDD15	A2, A32, B1, B33, C3, C7, C11, C17, C21, C25, C31, D5, D9, D14, D19, D23, D27, D29, F5, F7, F9, F11, F13, F17, F19, F21, F23, F25, F27, F29, G10, G14, G20, G26, G28, G30, H7, H9, H11, H13, H15, H17, H19, H21, H23, H25, H27	1.5 V/1.35 V	DDR IO supply
DVDD18	J2, J28, K27, L8, L28, M2, M7, M27, N8, N28, N31, P7, P27, R28, T7, T27, U28, U31, V7, V27, W28, Y7, Y27, AA3, AA8, AA28, AA31, AB7, AB27, AC8, AC28, AD4, AD27, AE8, AE26, AE28, AE31, AF27, AG26, AH27, AJ30, AM33, AN32	1.8 V	1.8-V IO supply
USB0DVDD33	Y11	3.3 V	3.3-V USB0 high supply High-speed
USB0VP	U8	0.85 V	0.85-V USB0 PHY analog and digital Super-speed supply
USB0VPH	W10	3.3 V	3.3-V USB0 high supply Super-speed
USB0VPTX	V9	0.85 V	0.85-V USB0 PHY transmit supply
USB1DVDD33	P11	3.3 V	3.3-V USB1 high supply High-speed
USB1VP	R8	0.85 V	0.85-V USB1 PHY analog and digital Super-speed supply
USB1VPH	R10	3.3 V	3.3-V USB1 high supply Super-speed
USB1VPTX	T9	0.85 V	0.85-V USB1 PHY transmit supply
VDDAHV	AH11, AH13, AH15, AH17, AH19, AH21, AH23, AH25	1.8 V	1.8-V high analog supply
VDDALV	AE10, AE12, AE14, AE16, AE18, AE20, AE22, AE24, AF9, AF11, AF13, AF15, AF17, AF19, AF21, AF23, AF25, AG16, AG18, AG20, AG22, AG24	0.85 V	SerDes low voltage
VDDUSB0	U10, V11	0.85 V	USB0 PHY analog and digital High-speed supply
VDDUSB1	R12, T11	0.85 V	USB1 PHY analog and digital High-speed supply
VNWA1	P9	0.95 V	Fixed Nwell supply - connect to CVDD1
VNWA2	J24	0.95 V	Fixed Nwell supply - connect to CVDD1
VNWA3	AD23	0.95 V	Fixed Nwell supply - connect to CVDD1
VNWA4	AD9	0.95 V	Fixed Nwell supply - connect to CVDD1
VPP0	K9		Leave unconnected
VPP1	M9		Leave unconnected

Table 5-3. Terminal Functions — Power and Ground (continued)

SUPPLY	BALL NO.	VOLTS	DESCRIPTION
VSS	A1, A33, C5, C9, C14, C19, C23, C27, C29, D3, D7, D11, D17, D21, D25, D31, F4, F6, F8, F10, F12, F14, F16, F18, F20, F22, F24, F26, F28, F30, G7, G9, G11, G13, G15, G17, G19, G21, G23, G25, G27, G29, G31, H8, H10, H12, H14, H16, H18, H20, H22, H24, H26, H28, J3, J7, J11, J13, J15, J17, J19, J21, J23, J25, J27, K10, K12, K14, K16, K18, K20, K22, K24, K28, L7, L9, L11, L13, L15, L17, L19, L21, L23, L25, L27, M1, M8, M10, M12, M14, M16, M18, M20, M22, M24, M26, M28, N2, N7, N9, N11, N13, N15, N17, N19, N21, N23, N25, N27, N32, P8, , P10, P12, P14, P16, P18, P20, P22, P24, P26, P28, R1, R7, R9, R11, R13, R15, R17, R19, R21, R23, R25, R27, T2, T8, T10, T12, T14, T16, T18, T20, T22, T24, T26, T28, U3, U7, U9, U11, U13, U15, U17, U19, U21, U23, U25, U27, U32, V1, V8, V10, V12, V14, V16, V18, V20, V22, V24, V26, V28, W2, W9, W11, W13, W15, W17, W19, W21, W23, W25, W27, Y8, Y10, Y12, Y14, Y16, Y18, Y20, Y22, Y24, Y26, Y28, AA4, AA7, AA9, AA11, AA13, AA15, AA17, AA19, AA21, AA23, AA25, AA27, AA32, AB8, AB10, AB12, AB14, AB16, AB18, AB20, AB22, AB24, AB26, AB28, AC7, AC9, AC11, AC13, AC15, AC17, AC19, AC21, AC23, AC25, AC27, AD5, AD8, AD10, AD12, AD14, AD16, AD18, AD20, AD22, AD24, AD26, AD28, AE9, AE11, AE13, AE15, AE17, AE19, AE21, AE23, AE25, AE27, AE32, AF8, AF10, AF12, AF14, AF16, AF18, AF20, AF22, AF24, AF26, AF28, AG7, AG9, AG11, AG13, AG15, AG17, AG19, AG21, AG23, AG25, AG27, AH8, AH10, AH12, AH14, AH16, AH18, AH20, AH22, AH24, AH26, AH28, AJ4, AJ6, AJ9, AJ13, AJ16, AJ19, AJ21, AJ23, AJ29, AK3, AK6, AK9, AK12, AK15, AK18, AK21, AK24, AK27, AK30, AL4, AL7, AL10, AL13, AL16, AL19, AL22, AL25, AL28, AL31, AM1, AM3, AM6, AM9, AM12, AM15, AM18, AM21, AM24, AM27, AM30, AN1, AN2, AN4, AN7, AN10, AN13, AN16, AN19, AN22, AN25, AN28, AN31, AN33	GND	Ground
VSSCMON	J9	GND	GND Monitor
VSSTMON	K26	GND	GND Monitor

Table 5-4. Terminal Functions — By Signal Name

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
BOOTMODE_RSVD[2]	Y31	DDRA02	B16	DDRD15	B3
AVDDA1	AF7	DDRA03	C16	DDRD16	D6
AVDDA2	K7	DDRA04	D16	DDRD17	E6
AVDDA3	AD7	DDRA05	E16	DDRD18	E7
AVDDA6	G8	DDRA06	A17	DDRD19	B6
AVDDA7	E14	DDRA07	E17	DDRD20	C6
AVDDA8	G16	DDRA08	A18	DDRD21	C8
AVDDA9	G22	DDRA09	B17	DDRD22	A8
AVDDA10	G24	DDRA10	A13	DDRD23	B8
AVSIFSEL[0][2]	K33	DDRA11	D18	DDRD24	E10
AVSIFSEL[1][2]	K32	DDRA12	E18	DDRD25	B11
BOOTCOMPLETE	AF31	DDRA13	E12	DDRD26	A11
BOOTMODE00[2]	AA29	DDRA14	C18	DDRD27	C10
BOOTMODE01[2]	Y29	DDRA15	E19	DDRD28	D10
BOOTMODE02[2]	V33	DDRBA0	D13	DDRD29	B9
BOOTMODE03[2]	V32	DDRBA1	C13	DDRD30	E9
BOOTMODE04[2]	W30	DDRBA2	B18	DDRD31	A9
BOOTMODE05[2]	W32	$\overline{\text{DDRCAS}}$	B12	DDRD32	B23
BOOTMODE06[2]	V31	DDRCB00	A23	DDRD33	E23
BOOTMODE07[2]	W31	DDRCB01	D22	DDRD34	D24
BOOTMODE08[2]	W33	DDRCB02	E22	DDRD35	C24
BOOTMODE09[2]	AB29	DDRCB03	E21	DDRD36	E24
BOOTMODE10[2]	Y30	DDRCB04	C22	DDRD37	E25
BOOTMODE11[2]	Y32	DDRCB05	B21	DDRD38	B25
BOOTMODE12[2]	AA30	DDRCB06	A20	DDRD39	A25
BOOTMODE13[2]	AA33	DDRCB07	A21	DDRD40	E28
BOOTMODE14[2]	AB32	$\overline{\text{DDRCE0}}$	A12	DDRD41	D28
BOOTMODE15[2]	AB33	$\overline{\text{DDRCE1}}$	C12	DDRD42	C28
CORECLKN	AG1	DDRCKE0	C20	DDRD43	E27
CORECLKP	AF1	DDRCKE1	A19	DDRD44	B28
CVDD	J12, J14, J16, J18, J20, J22, J26, K11, K13, K15, K17, K19, K21, K23, K25, L10, L12, L14, L16, L18, L20, L22, M15, M17, M19, M21, M25, N10, N14, N16, N18, N20, N22	DDRCLKN	G3	DDRD45	A26
		DDRCLKOUTN0	B14	DDRD46	B26
		DDRCLKOUTN1	A15	DDRD47	C26
		DDRCLKOUTP0	B15	DDRD48	A29
CVDD	N26, P13, P15, P17, P19, P21, P23, P25, R14, R16, R18, R20, R22, R24, R26, T13, T15, T17, T19, T21, T23, T25, U12, U14, U16, U18, U20, U22, U24, U26, V13, V15	DDRCLKOUTP1	A16	DDRD49	A28
		DDRCLKP	G4	DDRD50	B31
		DDRD00	E2	DDRD51	E29
		DDRD01	D1	DDRD52	B29
CVDD	V17, V19, V21, V23, V25, W12, W14, W16, W18, W20, W22, W24, W26, Y9, Y15, Y17, Y19, Y21, Y25, AA10, AA16, AA18, AA20, AA22, AA26, AB9, AB11, AB13, AB15	DDRD02	F2	DDRD53	C30
		DDRD03	E1	DDRD54	E30
		DDRD04	F1	DDRD55	D30
		DDRD05	D2	DDRD56	F31
CVDD	AB17, AB19, AB21, AB25, AC10, AC12, AC14, AC16, AC18, AC20, AC22, AC24, AC26, AD11, AD13, AD15, AD17, AD19, AD21, AD25	DDRD06	F3	DDRD57	F33
		DDRD07	B2	DDRD58	F32
		DDRD08	A5	DDRD59	E32
CVDD1	L24, M11, M13, M23, N12, N24, Y13, Y23, AA12, AA14, AA24, AB23	DDRD09	A6	DDRD60	E33
		DDRD10	B5	DDRD61	C32
CVDDCMON	J10	DDRD11	E5	DDRD62	D32
CVDDTMON	L26	DDRD12	C4	DDRD63	B32
DDRA00	D15	DDRD13	D4	DDRDQM0	A3
DDRA01	C15	DDRD14	E4	DDRDQM1	E3

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
DDRDQM2	D8	EMIFA05	N6	EMU05	AD3
DDRDQM3	E8	EMIFA06	M3	EMU06	AB5
DDRDQM4	E26	EMIFA07	N4	EMU07	AC3
DDRDQM5	D26	EMIFA08	N5	EMU08	AB4
DDRDQM6	E31	EMIFA09	M5	EMU09	AH3
DDRDQM7	A31	EMIFA10	M4	EMU10	AF3
DDRDQM8	B20	EMIFA11	L1	EMU11	AG4
DDRDQS0N	C2	EMIFA12	U5	EMU12	AD2
DDRDQS0P	C1	EMIFA13	T5	EMU13	AC2
DDRDQS1N	B4	EMIFA14	L3	EMU14	AB3
DDRDQS1P	A4	EMIFA15	T3	EMU15	AA5
DDRDQS2N	A7	EMIFA16	L2	EMU16	AB2
DDRDQS2P	B7	EMIFA17	L5	EMU17	Y5
DDRDQS3N	B10	EMIFA18	M6	EMU18	AH2
DDRDQS3P	A10	EMIFA19	K2	EMU19[3]	AB32
DDRDQS4N	B24	EMIFA20	K1	EMU20[3]	AB33
DDRDQS4P	A24	EMIFA21	L4	EMU21[3]	AB31
DDRDQS5N	B27	EMIFA22	R3	EMU22[3]	AC29
DDRDQS5P	A27	EMIFA23	L6	EMU23[3]	AC33
DDRDQS6N	B30	$\overline{\text{EMIFBE0}}$	AA2	EMU24[3]	AD29
DDRDQS6P	A30	$\overline{\text{EMIFBE1}}$	R6	EMU25[3]	AC31
DDRDQS7N	D33	$\overline{\text{EMIFCE0}}$	AA1	EMU26[3]	AC32
DDRDQS7P	C33	$\overline{\text{EMIFCE1}}$	Y4	EMU27[3]	AB30
DDRDQS8N	A22	$\overline{\text{EMIFCE2}}$	W1	EMU28[3]	AC30
DDRDQS8P	B22	$\overline{\text{EMIFCE3}}$	Y1	EMU29[3]	AD32
DDRODT0	D12	EMIFD00	G5	EMU30[3]	AD33
DDRODT1	E11	EMIFD01	K3	EMU31[3]	AD31
$\overline{\text{DDRRAS}}$	B13	EMIFD02	K4	EMU32[3]	AE33
$\overline{\text{DDRRESET}}$	D20	EMIFD03	J1	EMU33[3]	AD30
DDRRZQ0	E15	EMIFD04	H5	GPIO00	V30
DDRRZQ1	G12	EMIFD05	J6	GPIO01	AA29
DDRRZQ2	G18	EMIFD06	H6	GPIO02	Y29
DDRVREFSSTL	F15	EMIFD07	H1	GPIO03	V33
$\overline{\text{DDRWE}}$	E13	EMIFD08	H2	GPIO04	V32
DVDD15	A2, A32, B1, B33, C3, C7, C11, C17, C21, C25, C31, D5, D9, D14, D19, D23, D27, D29, F5, F7, F9, F11, F13, F17, F19, F21	EMIFD09	G6	GPIO05	W30
		EMIFD10	K5	GPIO06	W32
		EMIFD11	H3	GPIO07	V31
DVDD15	F23, F25, F27, F29, G10, G14, G20, G26, G28, G30, H7, H9, H11, H13, H15, H17, H19, H21, H23, H25, H27	EMIFD12	J4	GPIO08	W31
		EMIFD13	H4	GPIO09	W33
		EMIFD14	K6	GPIO10	AB29
DVDD18	J2, J28, K27, L8, L28, M2, M7, M27, N8, N28, N31, P7, P27, R28, T7, T27, U28, U31, V7, V27, W28, Y7, Y27, AA3, AA8	EMIFD15	J5	GPIO11	Y30
		$\overline{\text{EMIFOE}}$	Y2	GPIO12	Y32
		EMIFR$\overline{\text{W}}$	Y3	GPIO13	AA30
DVDD18	AA28, AA31, AB7, AB27, AC8, AC28, AD4, AD27, AE8, AE26, AE28, AE31, AF27, AG26, AH27, AJ30, AM33, AN32	EMIFWAIT0	T6	GPIO14	Y33
		EMIFWAIT1	N3	GPIO15	Y31
		$\overline{\text{EMIFWE}}$	R5	GPIO16	AA33
EMIFA00	P3	EMU00	AC5	GPIO17	AB32
EMIFA01	P5	EMU01	AA6	GPIO18	AB33
EMIFA02	U6	EMU02	AC4	GPIO19	AB31
EMIFA03	V6	EMU03	AE6	GPIO20	AC29
EMIFA04	P6	EMU04	AD6	GPIO21	AC33

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
GPIO22	AD29	PCIE0RXP1	AM17	SDA1	AJ32
GPIO23	AC31	PCIE0TXN0	AL18	SDA2	AG29
GPIO24	AC32	PCIE0TXN1	AK17	SGMII00REFRES	AJ27
GPIO25	AB30	PCIE0TXP0	AL17	SGMII01REFRES	AJ22
GPIO26	AC30	PCIE0TXP1	AK16	SGMII0CLKN	AJ25
GPIO27	AD32	PCIE1CLKN	AJ15	SGMII0CLKP	AJ24
GPIO28	AD33	PCIE1CLKP	AJ14	SGMII0RXN0	AN29
GPIO29	AD31	PCIE1REFRES	AJ10	SGMII0RXN1	AM28
GPIO30	AE33	PCIE1RXN0	AN14	SGMII0RXN2	AN26
GPIO31	AD30	PCIE1RXN1	AM13	SGMII0RXN3	AM25
HOUT	AF30	PCIE1RXP0	AN15	SGMII0RXN4	AN23
HYPLNK0CLKN	AJ8	PCIE1RXP1	AM14	SGMII0RXN5	AM22
HYPLNK0CLKP	AJ7	PCIE1TXN0	AL15	SGMII0RXN6	AN20
HYPLNK0REFRES	AJ5	PCIE1TXN1	AK14	SGMII0RXN7	AM19
HYPLNK0RXFLCLK	AJ3	PCIE1TXP0	AL14	SGMII0RXP0	AN30
HYPLNK0RXFLDAT	AE3	PCIE1TXP1	AK13	SGMII0RXP1	AM29
HYPLNK0RXN0	AN8	$\overline{\text{POR}}$	AH33	SGMII0RXP2	AN27
HYPLNK0RXN1	AM7	$\overline{\text{RESETFULL}}$	AF32	SGMII0RXP3	AM26
HYPLNK0RXN2	AN5	$\overline{\text{RESETSTAT}}$	AH29	SGMII0RXP4	AN24
HYPLNK0RXN3	AM4	$\overline{\text{RESET}}$	AE29	SGMII0RXP5	AM23
HYPLNK0RXP0	AN9	RSV000	H32	SGMII0RXP6	AN21
HYPLNK0RXP1	AM8	RSV001	J32	SGMII0RXP7	AM20
HYPLNK0RXP2	AN6	RSV002	AE2	SGMII0TXN0	AL30
HYPLNK0RXP3	AM5	RSV003	AF2	SGMII0TXN1	AK29
HYPLNK0RXPMCLK	AE1	RSV004	G2	SGMII0TXN2	AL27
HYPLNK0RXPMDAT	AD1	RSV005	G1	SGMII0TXN3	AK26
HYPLNK0TXFLCLK	AC6	RSV006	AL3	SGMII0TXN4	AL24
HYPLNK0TXFLDAT	AH4	RSV007	AL2	SGMII0TXN5	AK23
HYPLNK0TXN0	AL9	RSV008	AG5	SGMII0TXN6	AL21
HYPLNK0TXN1	AK8	RSV009	K8	SGMII0TXN7	AK20
HYPLNK0TXN2	AL6	RSV010	J8	SGMII0TXP0	AL29
HYPLNK0TXN3	AK5	RSV011	W8	SGMII0TXP1	AK28
HYPLNK0TXP0	AL8	RSV012	W7	SGMII0TXP2	AL26
HYPLNK0TXP1	AK7	RSV013	J30	SGMII0TXP3	AK25
HYPLNK0TXP2	AL5	RSV014	J29	SGMII0TXP4	AL23
HYPLNK0TXP3	AK4	RSV015	AG8	SGMII0TXP5	AK22
HYPLNK0TXPMCLK	AB6	RSV016	AJ20	SGMII0TXP6	AL20
HYPLNK0TXPMDAT	AF5	RSV017	AG12	SGMII0TXP7	AK19
LENDIAN[2]	V30	RSV018	AJ28	SPI0CLK	M30
MAINPLLODSEL[2]	Y33	RSV019	AJ26	SPI0DIN	M31
MDCLK0	AH6	RSV020	AH9	SPI0DOUT	N29
MDIO0	AH5	RSV021	B19	SPI0SCS0	L31
NETCPCLKN	AN3	RSV022	E20	SPI0SCS1	M29
NETCPCLKP	AM2	RSV023	A14	SPI0SCS2	L29
NETCPCLKSEL	AG6	RSV028	AG33	SPI0SCS3	L32
PCIE0CLKN	AJ18	RSV029	AF33	SPI1CLK	M33
PCIE0CLKP	AJ17	RSV030	AE30	SPI1DIN	R30
PCIE0REFRES	AG14	SCL0	AG30	SPI1DOUT	P32
PCIE0RXN0	AN17	SCL1	AH30	SPI1SCS0	P29
PCIE0RXN1	AM16	SCL2	AH31	SPI1SCS1	M32
PCIE0RXP0	AN18	SDA0	AG28	SPI1SCS2	N30

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
SPI1SCS3	N33	TSRXCLKOUT0P	AJ1	USB1RESREF	V5
SPI2CLK	L33	TSRXCLKOUT1N	AH1	USB1RX0M	N1
SPI2DIN	P30	TSRXCLKOUT1P	AG2	USB1RX0P	P1
SPI2DOUT	R31	TSSYNCEVT	AE5	USB1TX0M	R2
SPI2SCS0	P31	UART0CTS	R33	USB1TX0P	P2
SPI2SCS1	R29	UART0DSR	W29	USB1VBUS	V4
SPI2SCS2	P33	UART0DTR	U33	USB1VP	R8
SPI2SCS3	L30	UART0RTS	R32	USB1VPH	R10
SYSCLKOUT	AE4	UART0RXD	T29	USB1VPTX	T9
TCK	AJ33	UART0TXD	T30	USBCLKM	T4
TDI	AG31	UART1CTS	T31	USBCLKP	U4
TDO	AF29	UART1RTS	T33	USIMCLK	J33
TIMI0	K33	UART1RXD	U29	USIMIO	H33
TIMI1	K32	UART1TXD	T32	USIMRST	K29
TIMO0	K31	USB0DM	W3	VCL	V29
TIMO1	K30	USB0DP	V3	VCNTL0	H29
TMS	AH32	USB0DRVVBUS	AB1	VCNTL1	G33
$\overline{\text{TRST}}$	AG32	USB0DVDD33	Y11	VCNTL2	H31
TSCOMPOUT	AF4	USB0ID0	W4	VCNTL3	H30
TSIP0CLKA	AK31	USB0RESREF	Y6	VCNTL4	G32
TSIP0CLKB	AK33	USB0RX0M	T1	VCNTL5	J31
TSIP0FSA	AJ31	USB0RX0P	U1	VD	U30
TSIP0FSB	AK32	USB0TX0M	U2	VDDAHV	AH11, AH13, AH15, AH17, AH19, AH21, AH23, AH25
TSIP0TR0	AM31	USB0TX0P	V2		
TSIP0TR1	AL33	USB0VBUS	W5	VDDALV	AE10, AE12, AE14, AE16, AE18, AE20, AE22, AE24, AF9, AF11, AF13, AF15, AF17, AF19, AF21, AF23, AF25, AG16, AG18, AG20, AG22, AG24
TSIP0TX0	AL32	USB0VP	U8		
TSIP0TX1	AM32	USB0VPH	W10		
TSPUSHEVT0	AG3	USB0VPTX	V9		
TSPUSHEVT1	AL1	USB1DM	P4		
TSREFCLKN	AK1	USB1DP	R4	VDDUSB0	U10, V11
TSREFCLKP	AK2	USB1DRVVBUS	AC1	VDDUSB1	R12, T11
TSRXCLKOUT0N	AJ2	USB1DVDD33	P11	VNWA1	P9
		USB1ID0	W6	VNWA2	J24

SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER
VNWA3	AD23	VSS	W11, W13, W15, W17, W19, W21, W23, W25, W27, Y8, Y10, Y12, Y14, Y16, Y18, Y20, Y22, Y24, Y26, Y28, AA4, AA7, AA9, AA11, AA13, AA15, AA17, AA19, AA21, AA23	VSS	AN4, AN7, AN10, AN13, AN16, AN19, AN22, AN25, AN28, AN31, AN33
VNWA4	AD9				
VPP0	K9				
VPP1	M9			VSSCMON	J9
VSS	A1, A33, C5, C9, C14, C19, C23, C27, C29, D3, D7, D11, D17, D21, D25, D31, F4, F6, F8, F10, F12, F14, F16, F18, F20, F22, F24, F26, F28, F30, G7, G9, G11, G13, G15, G17	VSS	AA25, AA27, AA32, AB8, AB10, AB12, AB14, AB16, AB18, AB20, AB22, AB24, AB26, AB28, AC7, AC9, AC11, AC13, AC15, AC17, AC19, AC21, AC23, AC25	VSSTMON	K26
				XFICLKN	AJ12
				XFICLKP	AJ11
				XFIMDCLK	AF6
VSS	G19, G21, G23, G25, G27, G29, G31, H8, H10, H12, H14, H16, H18, H20, H22, H24, H26, H28, J3, J7, J11, J13, J15, J17, J19, J21, J23, J25, J27, K10, K12, K14, K16, K18	VSS	AC27, AD5, AD8, AD10, AD12, AD14, AD16, AD18, AD20, AD22, AD24, AD26, AD28, AE9, AE11, AE13, AE15, AE17, AE19, AE21, AE23, AE25, AE27, AE32	XFIMDIO	AE7
				XFIREFRES0	AG10
				XFIREFRES1	AH7
				XFIRXN0	AN11
VSS	K20, K22, K24, K28, L7, L9, L11, L13, L15, L17, L19, L21, L23, L25, L27, M1, M8, M10, M12, M14, M16, M18, M20, M22, M24, M26, M28, N2, N7, N9, N11, N13, N15, N17	VSS	AF8, AF10, AF12, AF14, AF16, AF18, AF20, AF22, AF24, AF26, AF28, AG7, AG9, AG11, AG13, AG15, AG17, AG19, AG21, AG23, AG25, AG27, AH8, AH10	XFIRXN1	AM10
				XFIRXP0	AN12
				XFIRXP1	AM11
				XFITXN0	AL12
VSS	N19, N21, N23, N25, N27, N32, P8, P10, P12, P14, P16, P18, P20, P22, P24, P26, P28, R1, R7, R9, R11, R13, R15, R17, R19, R21, R23, R25, R27, T2, T8, T10, T12, T14	VSS	AH12, AH14, AH16, AH18, AH20, AH22, AH24, AH26, AH28, AJ4, AJ6, AJ9, AJ13, AJ16, AJ19, AJ21, AJ23, AJ29, AK3, AK6, AK9, AK12, AK15, AK18, AK21, AK24	XFITXN1	AK11
				XFITXP0	AL11
				XFITXP1	AK10
VSS	T16, T18, T20, T22, T24, T26, T28, U3, U7, U9, U11, U13, U15, U17, U19, U21, U23, U25, U27, U32, V1, V8, V10, V12, V14, V16, V18, V20, V22, V24, V26, V28, W2, W9	VSS	AK27, AK30, AL4, AL7, AL10, AL13, AL16, AL19, AL22, AL25, AL28, AL31, AM1, AM3, AM6, AM9, AM12, AM15, AM18, AM21, AM24, AM27, AM30, AN1, AN2		

Table 5-5. Terminal Functions — By Ball Number

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
A1	VSS	B20	DDRDQM8	D6	DDRD16
A2	DVDD15	B21	DDRCB05	D7	VSS
A3	DDRDQM0	B22	DDRDQS8P	D8	DDRDQM2
A4	DDRDQS1P	B23	DDRD32	D9	DVDD15
A5	DDRD08	B24	DDRDQS4N	D10	DDRD28
A6	DDRD09	B25	DDRD38	D11	VSS
A7	DDRDQS2N	B26	DDRD46	D12	DDRODT0
A8	DDRD22	B27	DDRDQS5N	D13	DDRBA0
A9	DDRD31	B28	DDRD44	D14	DVDD15
A10	DDRDQS3P	B29	DDRD52	D15	DDRA00
A11	DDRD26	B30	DDRDQS6N	D16	DDRA04
A12	$\overline{\text{DDRCE0}}$	B31	DDRD50	D17	VSS
A13	DDRA10	B32	DDRD63	D18	DDRA11
A14	RSV023	B33	DVDD15	D19	DVDD15
A15	DDRCLKOUTN1	C1	DDRDQS0P	D20	$\overline{\text{DDRRESET}}$
A16	DDRCLKOUTP1	C2	DDRDQS0N	D21	VSS
A17	DDRA06	C3	DVDD15	D22	DDRCB01
A18	DDRA08	C4	DDRD12	D23	DVDD15
A19	DDRCKE1	C5	VSS	D24	DDRD34
A20	DDRCB06	C6	DDRD20	D25	VSS
A21	DDRCB07	C7	DVDD15	D26	DDRDQM5
A22	DDRDQS8N	C8	DDRD21	D27	DVDD15
A23	DDRCB00	C9	VSS	D28	DDRD41
A24	DDRDQS4P	C10	DDRD27	D29	DVDD15
A25	DDRD39	C11	DVDD15	D30	DDRD55
A26	DDRD45	C12	$\overline{\text{DDRCE1}}$	D31	VSS
A27	DDRDQS5P	C13	DDRBA1	D32	DDRD62
A28	DDRD49	C14	VSS	D33	DDRDQS7N
A29	DDRD48	C15	DDRA01	E1	DDRD03
A30	DDRDQS6P	C16	DDRA03	E2	DDRD00
A31	DDRDQM7	C17	DVDD15	E3	DDRDQM1
A32	DVDD15	C18	DDRA14	E4	DDRD14
A33	VSS	C19	VSS	E5	DDRD11
B1	DVDD15	C20	DDRCKE0	E6	DDRD17
B2	DDRD07	C21	DVDD15	E7	DDRD18
B3	DDRD15	C22	DDRCB04	E8	DDRDQM3
B4	DDRDQS1N	C23	VSS	E9	DDRD30
B5	DDRD10	C24	DDRD35	E10	DDRD24
B6	DDRD19	C25	DVDD15	E11	DDRODT1
B7	DDRDQS2P	C26	DDRD47	E12	DDRA13
B8	DDRD23	C27	VSS	E13	$\overline{\text{DDRWE}}$
B9	DDRD29	C28	DDRD42	E14	AVDDA7
B10	DDRDQS3N	C29	VSS	E15	DDRRZQ0
B11	DDRD25	C30	DDRD53	E16	DDRA05
B12	$\overline{\text{DDRCAS}}$	C31	DVDD15	E17	DDRA07
B13	$\overline{\text{DDRRAS}}$	C32	DDRD61	E18	DDRA12
B14	DDRCLKOUTN0	C33	DDRDQS7P	E19	DDRA15
B15	DDRCLKOUTP0	D1	DDRD01	E20	RSV022
B16	DDRA02	D2	DDRD05	E21	DDRCB03
B17	DDRA09	D3	VSS	E22	DDRCB02
B18	DDRBA2	D4	DDRD13	E23	DDRD33
B19	RSV021	D5	DVDD15	E24	DDRD36

Table 5-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
E25	DDRD37	G11	VSS	H30	VCNTL3
E26	DDRDQM4	G12	DDRRZQ1	H31	VCNTL2
E27	DDRD43	G13	VSS	H32	RSV000
E28	DDRD40	G14	DVDD15	H33	USIMIO
E29	DDRD51	G15	VSS	J1	EMIFD03
E30	DDRD54	G16	AVDDA8	J2	DVDD18
E31	DDRDQM6	G17	VSS	J3	VSS
E32	DDRD59	G18	DDRRZQ2	J4	EMIFD12
E33	DDRD60	G19	VSS	J5	EMIFD15
F1	DDRD04	G20	DVDD15	J6	EMIFD05
F2	DDRD02	G21	VSS	J7	VSS
F3	DDRD06	G22	AVDDA9	J8	RSV010
F4	VSS	G23	VSS	J9	VSSCMON
F5	DVDD15	G24	AVDDA10	J10	CVDDCMON
F6	VSS	G25	VSS	J11	VSS
F7	DVDD15	G26	DVDD15	J12	CVDD
F8	VSS	G27	VSS	J13	VSS
F9	DVDD15	G28	DVDD15	J14	CVDD
F10	VSS	G29	VSS	J15	VSS
F11	DVDD15	G30	DVDD15	J16	CVDD
F12	VSS	G31	VSS	J17	VSS
F13	DVDD15	G32	VCNTL4	J18	CVDD
F14	VSS	G33	VCNTL1	J19	VSS
F15	DDRVREFSSTL	H1	EMIFD07	J20	CVDD
F16	VSS	H2	EMIFD08	J21	VSS
F17	DVDD15	H3	EMIFD11	J22	CVDD
F18	VSS	H4	EMIFD13	J23	VSS
F19	DVDD15	H5	EMIFD04	J24	VNWA2
F20	VSS	H6	EMIFD06	J25	VSS
F21	DVDD15	H7	DVDD15	J26	CVDD
F22	VSS	H8	VSS	J27	VSS
F23	DVDD15	H9	DVDD15	J28	DVDD18
F24	VSS	H10	VSS	J29	RSV014
F25	DVDD15	H11	DVDD15	J30	RSV013
F26	VSS	H12	VSS	J31	VCNTL5
F27	DVDD15	H13	DVDD15	J32	RSV001
F28	VSS	H14	VSS	J33	USIMCLK
F29	DVDD15	H15	DVDD15	K1	EMIFA20
F30	VSS	H16	VSS	K2	EMIFA19
F31	DDRD56	H17	DVDD15	K3	EMIFD01
F32	DDRD58	H18	VSS	K4	EMIFD02
F33	DDRD57	H19	DVDD15	K5	EMIFD10
G1	RSV005	H20	VSS	K6	EMIFD14
G2	RSV004	H21	DVDD15	K7	AVDDA2
G3	DDRCLKN	H22	VSS	K8	RSV009
G4	DDRCLKP	H23	DVDD15	K9	VPP0
G5	EMIFD00	H24	VSS	K10	VSS
G6	EMIFD09	H25	DVDD15	K11	CVDD
G7	VSS	H26	VSS	K12	VSS
G8	AVDDA6	H27	DVDD15	K13	CVDD
G9	VSS	H28	VSS	K14	VSS
G10	DVDD15	H29	VCNTL0	K15	CVDD

Table 5-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
K16	VSS	L33	SPI2CLK	N19	VSS
K17	CVDD	M1	VSS	N20	CVDD
K18	VSS	M2	DVDD18	N21	VSS
K19	CVDD	M3	EMIFA06	N22	CVDD
K20	VSS	M4	EMIFA10	N23	VSS
K21	CVDD	M5	EMIFA09	N24	CVDD1
K22	VSS	M6	EMIFA18	N25	VSS
K23	CVDD	M7	DVDD18	N26	CVDD
K24	VSS	M8	VSS	N27	VSS
K25	CVDD	M9	VPP1	N28	DVDD18
K26	VSSTMON	M10	VSS	N29	SPI0DOUT
K27	DVDD18	M11	CVDD1	N30	SPI1SCS2
K28	VSS	M12	VSS	N31	DVDD18
K29	USIMRST	M13	CVDD1	N32	VSS
K30	TIMO1	M14	VSS	N33	SPI1SCS3
K31	TIMO0	M15	CVDD	P1	USB1RX0P
K32	TIMI1	M16	VSS	P2	USB1TX0P
K32	AVSIFSEL[1][2]	M17	CVDD	P3	EMIFA00
K33	TIMI0	M18	VSS	P4	USB1DM
K33	AVSIFSEL[0][2]	M19	CVDD	P5	EMIFA01
L1	EMIFA11	M20	VSS	P6	EMIFA04
L2	EMIFA16	M21	CVDD	P7	DVDD18
L3	EMIFA14	M22	VSS	P8	VSS
L4	EMIFA21	M23	CVDD1	P9	VNWA1
L5	EMIFA17	M24	VSS	P10	VSS
L6	EMIFA23	M25	CVDD	P11	USB1DVDD33
L7	VSS	M26	VSS	P12	VSS
L8	DVDD18	M27	DVDD18	P13	CVDD
L9	VSS	M28	VSS	P14	VSS
L10	CVDD	M29	SPI0SCS1	P15	CVDD
L11	VSS	M30	SPI0CLK	P16	VSS
L12	CVDD	M31	SPI0DIN	P17	CVDD
L13	VSS	M32	SPI1SCS1	P18	VSS
L14	CVDD	M33	SPI1CLK	P19	CVDD
L15	VSS	N1	USB1RX0M	P20	VSS
L16	CVDD	N2	VSS	P21	CVDD
L17	VSS	N3	EMIFWAIT1	P22	VSS
L18	CVDD	N4	EMIFA07	P23	CVDD
L19	VSS	N5	EMIFA08	P24	VSS
L20	CVDD	N6	EMIFA05	P25	CVDD
L21	VSS	N7	VSS	P26	VSS
L22	CVDD	N8	DVDD18	P27	DVDD18
L23	VSS	N9	VSS	P28	VSS
L24	CVDD1	N10	CVDD	P29	SPI1SCS0
L25	VSS	N11	VSS	P30	SPI2DIN
L26	CVDDTMON	N12	CVDD1	P31	SPI2SCS0
L27	VSS	N13	VSS	P32	SPI1DOUT
L28	DVDD18	N14	CVDD	P33	SPI2SCS2
L29	SPI0SCS2	N15	VSS	R1	VSS
L30	SPI2SCS3	N16	CVDD	R2	USB1TX0M
L31	SPI0SCS0	N17	VSS	R3	EMIFA22
L32	SPI0SCS3	N18	CVDD	R4	USB1DP

Table 5-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
R5	EMIFWE	T24	VSS	V10	VSS
R6	EMIFBE1	T25	CVDD	V11	VDDUSB0
R7	VSS	T26	VSS	V12	VSS
R8	USB1VP	T27	DVDD18	V13	CVDD
R9	VSS	T28	VSS	V14	VSS
R10	USB1VPH	T29	UART0RXD	V15	CVDD
R11	VSS	T30	UART0TXD	V16	VSS
R12	VDDUSB1	T31	UART1CTS	V17	CVDD
R13	VSS	T32	UART1TXD	V18	VSS
R14	CVDD	T33	UART1RTS	V19	CVDD
R15	VSS	U1	USB0RX0P	V20	VSS
R16	CVDD	U2	USB0TX0M	V21	CVDD
R17	VSS	U3	VSS	V22	VSS
R18	CVDD	U4	USBCLKP	V23	CVDD
R19	VSS	U5	EMIFA12	V24	VSS
R20	CVDD	U6	EMIFA02	V25	CVDD
R21	VSS	U7	VSS	V26	VSS
R22	CVDD	U8	USB0VP	V27	DVDD18
R23	VSS	U9	VSS	V28	VSS
R24	CVDD	U10	VDDUSB0	V29	VCL
R25	VSS	U11	VSS	V30	GPIO00
R26	CVDD	U12	CVDD	V30	LENDIAN[2]
R27	VSS	U13	VSS	V31	GPIO07
R28	DVDD18	U14	CVDD	V31	BOOTMODE06[2]
R29	SPI2SCS1	U15	VSS	V32	GPIO04
R30	SPI1DIN	U16	CVDD	V32	BOOTMODE03[2]
R31	SPI2DOUT	U17	VSS	V33	GPIO03
R32	UART0RTS	U18	CVDD	V33	BOOTMODE02[2]
R33	UART0CTS	U19	VSS	W1	EMIFCE2
T1	USB0RX0M	U20	CVDD	W2	VSS
T2	VSS	U21	VSS	W3	USB0DM
T3	EMIFA15	U22	CVDD	W4	USB0ID0
T4	USBCLKM	U23	VSS	W5	USB0VBUS
T5	EMIFA13	U24	CVDD	W6	USB1ID0
T6	EMIFWAIT0	U25	VSS	W7	RSV012
T7	DVDD18	U26	CVDD	W8	RSV011
T8	VSS	U27	VSS	W9	VSS
T9	USB1VPTX	U28	DVDD18	W10	USB0VPH
T10	VSS	U29	UART1RXD	W11	VSS
T11	VDDUSB1	U30	VD	W12	CVDD
T12	VSS	U31	DVDD18	W13	VSS
T13	CVDD	U32	VSS	W14	CVDD
T14	VSS	U33	UART0DTR	W15	VSS
T15	CVDD	V1	VSS	W16	CVDD
T16	VSS	V2	USB0TX0P	W17	VSS
T17	CVDD	V3	USB0DP	W18	CVDD
T18	VSS	V4	USB1VBUS	W19	VSS
T19	CVDD	V5	USB1RESREF	W20	CVDD
T20	VSS	V6	EMIFA03	W21	VSS
T21	CVDD	V7	DVDD18	W22	CVDD
T22	VSS	V8	VSS	W23	VSS
T23	CVDD	V9	USB0VPTX	W24	CVDD

Table 5-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
W25	VSS	AA2	EMIFBE0	AB18	VSS
W26	CVDD	AA3	DVDD18	AB19	CVDD
W27	VSS	AA4	VSS	AB20	VSS
W28	DVDD18	AA5	EMU15	AB21	CVDD
W29	UART0DSR	AA6	EMU01	AB22	VSS
W30	GPIO05	AA7	VSS	AB23	CVDD1
W30	BOOTMODE04[2]	AA8	DVDD18	AB24	VSS
W31	GPIO08	AA9	VSS	AB25	CVDD
W31	BOOTMODE07[2]	AA10	CVDD	AB26	VSS
W32	GPIO06	AA11	VSS	AB27	DVDD18
W32	BOOTMODE05[2]	AA12	CVDD1	AB28	VSS
W33	GPIO09	AA13	VSS	AB29	GPIO10
W33	BOOTMODE08[2]	AA14	CVDD1	AB29	BOOTMODE09[2]
Y1	EMIFCE3	AA15	VSS	AB30	GPIO25
Y2	EMIFOE	AA16	CVDD	AB30	EMU27[3]
Y3	EMIFRW	AA17	VSS	AB31	GPIO19
Y4	EMIFCE1	AA18	CVDD	AB31	EMU21[3]
Y5	EMU17	AA19	VSS	AB32	GPIO17
Y6	USB0RESREF	AA20	CVDD	AB32	EMU19[3]
Y7	DVDD18	AA21	VSS	AB32	BOOTMODE14[2]
Y8	VSS	AA22	CVDD	AB33	GPIO18
Y9	CVDD	AA23	VSS	AB33	EMU20[3]
Y10	VSS	AA24	CVDD1	AB33	BOOTMODE15[2]
Y11	USB0DVDD33	AA25	VSS	AC1	USB1DRVVBUS
Y12	VSS	AA26	CVDD	AC2	EMU13
Y13	CVDD1	AA27	VSS	AC3	EMU07
Y14	VSS	AA28	DVDD18	AC4	EMU02
Y15	CVDD	AA29	GPIO01	AC5	EMU00
Y16	VSS	AA29	BOOTMODE00[2]	AC6	HYPLNK0TXFLCLK
Y17	CVDD	AA30	GPIO13	AC7	VSS
Y18	VSS	AA30	BOOTMODE12[2]	AC8	DVDD18
Y19	CVDD	AA31	DVDD18	AC9	VSS
Y20	VSS	AA32	VSS	AC10	CVDD
Y21	CVDD	AA33	GPIO16	AC11	VSS
Y22	VSS	AA33	BOOTMODE13[2]	AC12	CVDD
Y23	CVDD1	AB1	USB0DRVVBUS	AC13	VSS
Y24	VSS	AB2	EMU16	AC14	CVDD
Y25	CVDD	AB3	EMU14	AC15	VSS
Y26	VSS	AB4	EMU08	AC16	CVDD
Y27	DVDD18	AB5	EMU06	AC17	VSS
Y28	VSS	AB6	HYPLNK0TXPMCLK	AC18	CVDD
Y29	GPIO02	AB7	DVDD18	AC19	VSS
Y29	BOOTMODE01[2]	AB8	VSS	AC20	CVDD
Y30	GPIO11	AB9	CVDD	AC21	VSS
Y30	BOOTMODE10[2]	AB10	VSS	AC22	CVDD
Y31	GPIO15	AB11	CVDD	AC23	VSS
Y31	BOOTMODE_RSVD[2]	AB12	VSS	AC24	CVDD
Y32	GPIO12	AB13	CVDD	AC25	VSS
Y32	BOOTMODE11[2]	AB14	VSS	AC26	CVDD
Y33	GPIO14	AB15	CVDD	AC27	VSS
Y33	MAINPLLODSEL[2]	AB16	VSS	AC28	DVDD18
AA1	EMIFCE0	AB17	CVDD	AC29	GPIO20

Table 5-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
AC29	EMU22[3]	AE6	EMU03	AF24	VSS
AC30	GPIO26	AE7	XFIMDIO	AF25	VDDALV
AC30	EMU28[3]	AE8	DVDD18	AF26	VSS
AC31	GPIO23	AE9	VSS	AF27	DVDD18
AC31	EMU25[3]	AE10	VDDALV	AF28	VSS
AC32	GPIO24	AE11	VSS	AF29	TDO
AC32	EMU26[3]	AE12	VDDALV	AF30	HOUT
AC33	GPIO21	AE13	VSS	AF31	BOOTCOMPLETE
AC33	EMU23[3]	AE14	VDDALV	AF32	$\overline{\text{RESETFULL}}$
AD1	HYPLNK0RXPMDAT	AE15	VSS	AF33	RSV029
AD2	EMU12	AE16	VDDALV	AG1	CORECLKN
AD3	EMU05	AE17	VSS	AG2	TSRXCLKOUT1P
AD4	DVDD18	AE18	VDDALV	AG3	TSPUSHEVT0
AD5	VSS	AE19	VSS	AG4	EMU11
AD6	EMU04	AE20	VDDALV	AG5	RSV008
AD7	AVDDA3	AE21	VSS	AG6	NETCPCLKSEL
AD8	VSS	AE22	VDDALV	AG7	VSS
AD9	VNWA4	AE23	VSS	AG8	RSV015
AD10	VSS	AE24	VDDALV	AG9	VSS
AD11	CVDD	AE25	VSS	AG10	XFIREFRES0
AD12	VSS	AE26	DVDD18	AG11	VSS
AD13	CVDD	AE27	VSS	AG12	RSV017
AD14	VSS	AE28	DVDD18	AG13	VSS
AD15	CVDD	AE29	$\overline{\text{RESET}}$	AG14	PCIE0REFRES
AD16	VSS	AE30	RSV030	AG15	VSS
AD17	CVDD	AE31	DVDD18	AG16	VDDALV
AD18	VSS	AE32	VSS	AG17	VSS
AD19	CVDD	AE33	GPIO30	AG18	VDDALV
AD20	VSS	AE33	EMU32[3]	AG19	VSS
AD21	CVDD	AF1	CORECLKP	AG20	VDDALV
AD22	VSS	AF2	RSV003	AG21	VSS
AD23	VNWA3	AF3	EMU10	AG22	VDDALV
AD24	VSS	AF4	TSCOMPOUT	AG23	VSS
AD25	CVDD	AF5	HYPLNK0TXPMDAT	AG24	VDDALV
AD26	VSS	AF6	XFIMDCLK	AG25	VSS
AD27	DVDD18	AF7	AVDDA1	AG26	DVDD18
AD28	VSS	AF8	VSS	AG27	VSS
AD29	GPIO22	AF9	VDDALV	AG28	SDA0
AD29	EMU24[3]	AF10	VSS	AG29	SDA2
AD30	GPIO31	AF11	VDDALV	AG30	SCL0
AD30	EMU33[3]	AF12	VSS	AG31	TDI
AD31	GPIO29	AF13	VDDALV	AG32	$\overline{\text{TRST}}$
AD31	EMU31[3]	AF14	VSS	AG33	RSV028
AD32	GPIO27	AF15	VDDALV	AH1	TSRXCLKOUT1N
AD32	EMU29[3]	AF16	VSS	AH2	EMU18
AD33	GPIO28	AF17	VDDALV	AH3	EMU09
AD33	EMU30[3]	AF18	VSS	AH4	HYPLNK0TXFLDAT
AE1	HYPLNK0RXPMCLK	AF19	VDDALV	AH5	MDIO0
AE2	RSV002	AF20	VSS	AH6	MDCLK0
AE3	HYPLNK0RXFLDAT	AF21	VDDALV	AH7	XFIREFRES1
AE4	SYSCLKOUT	AF22	VSS	AH8	VSS
AE5	TSSYNCEVT	AF23	VDDALV	AH9	RSV020

Table 5-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
AH10	VSS	AJ29	VSS	AL15	PCIE1TXN0
AH11	VDDAHV	AJ30	DVDD18	AL16	VSS
AH12	VSS	AJ31	TSIP0FSA	AL17	PCIE0TXP0
AH13	VDDAHV	AJ32	SDA1	AL18	PCIE0TXN0
AH14	VSS	AJ33	TCK	AL19	VSS
AH15	VDDAHV	AK1	TSREFCLKN	AL20	SGMII0TXP6
AH16	VSS	AK2	TSREFCLKP	AL21	SGMII0TXN6
AH17	VDDAHV	AK3	VSS	AL22	VSS
AH18	VSS	AK4	HYPLNK0TXP3	AL23	SGMII0TXP4
AH19	VDDAHV	AK5	HYPLNK0TXN3	AL24	SGMII0TXN4
AH20	VSS	AK6	VSS	AL25	VSS
AH21	VDDAHV	AK7	HYPLNK0TXP1	AL26	SGMII0TXP2
AH22	VSS	AK8	HYPLNK0TXN1	AL27	SGMII0TXN2
AH23	VDDAHV	AK9	VSS	AL28	VSS
AH24	VSS	AK10	XFITXP1	AL29	SGMII0TXP0
AH25	VDDAHV	AK11	XFITXN1	AL30	SGMII0TXN0
AH26	VSS	AK12	VSS	AL31	VSS
AH27	DVDD18	AK13	PCIE1TXP1	AL32	TSIP0TX0
AH28	VSS	AK14	PCIE1TXN1	AL33	TSIP0TR1
AH29	$\overline{\text{RESETSTAT}}$	AK15	VSS	AM1	VSS
AH30	SCL1	AK16	PCIE0TXP1	AM2	NETCPCLKP
AH31	SCL2	AK17	PCIE0TXN1	AM3	VSS
AH32	TMS	AK18	VSS	AM4	HYPLNK0RXN3
AH33	$\overline{\text{POR}}$	AK19	SGMII0TXP7	AM5	HYPLNK0RXP3
AJ1	TSRXCLKOUT0P	AK20	SGMII0TXN7	AM6	VSS
AJ2	TSRXCLKOUT0N	AK21	VSS	AM7	HYPLNK0RXN1
AJ3	HYPLNK0RXFLCLK	AK22	SGMII0TXP5	AM8	HYPLNK0RXP1
AJ4	VSS	AK23	SGMII0TXN5	AM9	VSS
AJ5	HYPLNK0REFRES	AK24	VSS	AM10	XFIRXN1
AJ6	VSS	AK25	SGMII0TXP3	AM11	XFIRXP1
AJ7	HYPLNK0CLKP	AK26	SGMII0TXN3	AM12	VSS
AJ8	HYPLNK0CLKN	AK27	VSS	AM13	PCIE1RXN1
AJ9	VSS	AK28	SGMII0TXP1	AM14	PCIE1RXP1
AJ10	PCIE1REFRES	AK29	SGMII0TXN1	AM15	VSS
AJ11	XFICLKP	AK30	VSS	AM16	PCIE0RXN1
AJ12	XFICLKN	AK31	TSIP0CLKA	AM17	PCIE0RXP1
AJ13	VSS	AK32	TSIP0FSB	AM18	VSS
AJ14	PCIE1CLKP	AK33	TSIP0CLKB	AM19	SGMII0RXN7
AJ15	PCIE1CLKN	AL1	TSPUSHEVT1	AM20	SGMII0RXP7
AJ16	VSS	AL2	RSV007	AM21	VSS
AJ17	PCIE0CLKP	AL3	RSV006	AM22	SGMII0RXN5
AJ18	PCIE0CLKN	AL4	VSS	AM23	SGMII0RXP5
AJ19	VSS	AL5	HYPLNK0TXP2	AM24	VSS
AJ20	RSV016	AL6	HYPLNK0TXN2	AM25	SGMII0RXN3
AJ21	VSS	AL7	VSS	AM26	SGMII0RXP3
AJ22	SGMII01REFRES	AL8	HYPLNK0TXP0	AM27	VSS
AJ23	VSS	AL9	HYPLNK0TXN0	AM28	SGMII0RXN1
AJ24	SGMII0CLKP	AL10	VSS	AM29	SGMII0RXP1
AJ25	SGMII0CLKN	AL11	XFITXP0	AM30	VSS
AJ26	RSV019	AL12	XFITXN0	AM31	TSIP0TR0
AJ27	SGMII00REFRES	AL13	VSS	AM32	TSIP0TX1
AJ28	RSV018	AL14	PCIE1TXP0	AM33	DVDD18

Table 5-5. Terminal Functions — By Ball Number (continued)

BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME	BALL NUMBER	SIGNAL NAME
AN1	VSS	AN12	XFIRXP0	AN23	SGMII0RXN4
AN2	VSS	AN13	VSS	AN24	SGMII0RXP4
AN3	NETCPCLKN	AN14	PCIE1RXN0	AN25	VSS
AN4	VSS	AN15	PCIE1RXP0	AN26	SGMII0RXN2
AN5	HYPLNK0RXN2	AN16	VSS	AN27	SGMII0RXP2
AN6	HYPLNK0RXP2	AN17	PCIE0RXN0	AN28	VSS
AN7	VSS	AN18	PCIE0RXP0	AN29	SGMII0RXN0
AN8	HYPLNK0RXN0	AN19	VSS	AN30	SGMII0RXP0
AN9	HYPLNK0RXP0	AN20	SGMII0RXN6	AN31	VSS
AN10	VSS	AN21	SGMII0RXP6	AN32	DVDD18
AN11	XFIRXN0	AN22	VSS	AN33	VSS

5.4 Pullup/Pulldown Resistors

Proper board design should ensure that input pins to the device always be at a valid logic level and not floating. This may be achieved via pullup/pulldown resistors. The device features internal pullup (IPU) and internal pulldown (IPD) resistors on most pins to eliminate the need, unless otherwise noted, for external pullup/pulldown resistors.

An external pullup/pulldown resistor needs to be used in the following situations:

- **Device Configuration Pins:** If the pin is both routed out and not driven (in Hi-Z state), an external pullup/pulldown resistor must be used, even if the IPU/IPD matches the desired value/state.
- **Other Input Pins:** If the IPU/IPD does not match the desired value/state, use an external pullup/pulldown resistor to pull the signal to the opposite rail.

For the device configuration pins (listed in Table 8-25), if they are both routed out and are not driven (in Hi-Z state), it is strongly recommended that an external pullup/pulldown resistor be implemented. Although, internal pullup/pulldown resistors exist on these pins and they may match the desired configuration value, providing external connectivity can help ensure that valid logic levels are latched on these device configuration pins. In addition, applying external pullup/pulldown resistors on the device configuration pins adds convenience to the user in debugging and flexibility in switching operating modes.

Tips for choosing an external pullup/pulldown resistor:

- Consider the total amount of current that may pass through the pullup or pulldown resistor. Be sure to include the leakage currents of all the devices connected to the net, as well as any internal pullup or pulldown resistors.
- Decide a target value for the net. For a pulldown resistor, this should be below the lowest V_{IL} level of all inputs connected to the net. For a pullup resistor, this should be above the highest V_{IH} level of all inputs on the net. A reasonable choice would be to target the V_{OL} or V_{OH} levels for the logic family of the limiting device; which, by definition, have margin to the V_{IL} and V_{IH} levels.
- Select a pullup/pulldown resistor with the largest possible value that still ensures that the net will reach the target pulled value when maximum current from all devices on the net is flowing through the resistor. The current to be considered includes leakage current plus, any other internal and external pullup/pulldown resistors on the net.
- For bidirectional nets, there is an additional consideration that sets a lower limit on the resistance value of the external resistor. Verify that the resistance is small enough that the weakest output buffer can drive the net to the opposite logic level (including margin).
- Remember to include tolerances when selecting the resistor value.
- For pullup resistors, also remember to include tolerances on the DVDD rail.

For most systems:

- A 1-kΩ resistor can be used to oppose the IPU/IPD while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.

- A 20-kΩ resistor can be used to compliment the IPU/IPD on the device configuration pins while meeting the above criteria. Users should confirm this resistor value is correct for their specific application.

For more detailed information on input current (I_I), and the low-level/high-level input voltages (V_{IL} and V_{IH}) for the AM5K2E0x device, see Section 9.3. To determine which pins on the device include internal pullup/pulldown resistors, see Table 5-2.

6 Memory, Interrupts, and EDMA for AM5K2E0x

6.1 Memory Map SummaryAM5K2E0x

The following table shows the memory map address ranges of the device.

Table 6-1. Device Memory Map Summary AM5K2E0x

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 0000 0000	00 0003 FFFF	256K	ARM ROM	ARM ROM
00 0004 0000	00 007F FFFF	8M-256K	Reserved	Reserved
00 0080 0000	00 008F FFFF	1M	Reserved	Reserved
00 0090 0000	00 00DF FFFF	5M	Reserved	Reserved
00 00E0 0000	00 00E0 7FFF	32K	Reserved	Reserved
00 00E0 8000	00 00EF FFFF	1M-32K	Reserved	Reserved
00 00F0 0000	00 00F0 7FFF	32K	Reserved	Reserved
00 00F0 8000	00 00FF FFFF	1M-32K	Reserved	Reserved
00 0100 0000	00 0100 FFFF	64K	ARM AXI2VBUSM registers	Reserved
00 0101 0000	00 010F FFFF	1M-64K	Reserved	Reserved
00 0110 0000	00 0110 FFFF	64K	ARM STM Stimulus Ports	Reserved
00 0101 0000	00 01BF FFFF	11M-64K	Reserved	Reserved
00 01C0 0000	00 01CF FFFF	1M	Reserved	Reserved
00 01D0 0000	00 01D0 007F	128	Tracer CFG0	Tracer CFG0
00 01D0 0080	00 01D0 7FFF	32K-128	Reserved	Reserved
00 01D0 8000	00 01D0 807F	128	Tracer CFG1	Tracer CFG1
00 01D0 8080	00 01D0 FFFF	32K-128	Reserved	Reserved
00 01D1 0000	00 01D1 007F	128	Tracer CFG2	Tracer CFG2
00 01D1 0080	00 01D1 7FFF	32K-128	Reserved	Reserved
00 01D1 8000	00 01D1 807F	128	Tracer CFG3	Tracer CFG3
00 01D1 8080	00 01D1 FFFF	32K-128	Reserved	Reserved
00 01D2 0000	00 01D2 007F	128	Tracer CFG4	Tracer CFG4
00 01D2 0080	00 01D2 7FFF	32K-128	Reserved	Reserved
00 01D2 8000	00 01D2 807F	128	Tracer CFG5	Tracer CFG5
00 01D2 8080	00 01D2 FFFF	32K-128	Reserved	Reserved
00 01D3 0000	00 01D3 007F	128	Tracer CFG6	Tracer CFG6
00 01D3 0080	00 01D3 7FFF	32K-128	Reserved	Reserved
00 01D3 8000	00 01D3 807F	128	Tracer CFG7	Tracer CFG7
00 01D3 8080	00 01D3 FFFF	32K-128	Reserved	Reserved
00 01D4 0000	00 01D4 007F	128	Tracer CFG8	Tracer CFG8
00 01D4 0080	00 01D4 7FFF	32K-128	Reserved	Reserved
00 01D4 8000	00 01D4 807F	128	Tracer CFG9	Tracer CFG9
00 01D4 8080	00 01D4 FFFF	32K-128	Reserved	Reserved
00 01D5 0000	00 01D5 007F	128	Reserved	Reserved
00 01D5 0080	00 01D5 7FFF	32K-128	Reserved	Reserved
00 01D5 8000	00 01D5 807F	128	Reserved	Reserved
00 01D5 8080	00 01D5 FFFF	32K-128	Reserved	Reserved
00 01D6 0000	00 01D6 007F	128	Reserved	Reserved
00 01D6 0080	00 01D6 7FFF	32K-128	Reserved	Reserved
00 01D6 8000	00 01D6 807F	128	Reserved	Reserved
00 01D6 8080	00 01D6 FFFF	32K-128	Reserved	Reserved
00 01D7 0000	00 01D7 007F	128	Reserved	Reserved
00 01D7 0080	00 01D7 7FFF	32K-128	Reserved	Reserved
00 01D7 8000	00 01D7 807F	128	Reserved	Reserved
00 01D7 8080	00 01D7 FFFF	32K-128	Reserved	Reserved

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 01D8 0000	00 01D8 007F	128	Reserved	Reserved
00 01D8 0080	00 01D8 7FFF	32K-128	Reserved	Reserved
00 01D8 8000	00 01D8 807F	128	Reserved	Reserved
00 01D8 8080	00 01D8 8FFF	32K-128	Reserved	Reserved
00 01D9 0000	00 01D9 007F	128	Reserved	Reserved
00 01D9 0080	00 01D9 7FFF	32K-128	Reserved	Reserved
00 01D9 8000	00 01D9 807F	128	Reserved	Reserved
00 01D9 8080	00 01D9 FFFF	32K-128	Reserved	Reserved
00 01DA 0000	00 01DA 007F	128	Tracer CFG20	Tracer CFG20
00 01DA 0080	00 01DA 7FFF	32K-128	Reserved	Reserved
00 01DA 8000	00 01DA 807F	128	Reserved	Reserved
00 01DA 8080	00 01DA FFFF	32K-128	Reserved	Reserved
00 01DB 0000	00 01DB 007F	128	Tracer CFG22	Tracer CFG22
00 01DB 0080	00 01DB 7FFF	32K-128	Reserved	Reserved
00 01DB 8000	00 01DB 807F	128	Reserved	Reserved
00 01DB 8080	00 01DB 8FFF	32K-128	Reserved	Reserved
00 01DC 0000	00 01DC 007F	128	Tracer CFG24	Tracer CFG24
00 01DC 0080	00 01DC 7FFF	32K-128	Reserved	Reserved
00 01DC 8000	00 01DC 807F	128	Tracer CFG25	Tracer CFG25
00 01DC 8080	00 01DC FFFF	32K-128	Reserved	Reserved
00 01DD 0000	00 01DD 007F	128	Tracer CFG26	Tracer CFG26
00 01DD 0080	00 01DD 7FFF	32K-128	Reserved	Reserved
00 01DD 8000	00 01DD 807F	128	Tracer CFG27	Tracer CFG27
00 01DD 8080	00 01DD FFFF	32K-128	Reserved	Reserved
00 01DE 0000	00 01DE 007F	128	Tracer CFG28	Tracer CFG28
00 01DE 0080	00 01DE 03FF	1K-128	Reserved	Reserved
00 01DE 0400	00 01DE 047F	128	Tracer CFG29	Tracer CFG29
00 01DD 0480	00 01DD 07FF	1K-128	Reserved	Reserved
00 01DE 0800	00 01DE 087F	128	Tracer CFG30	Tracer CFG30
00 01DE 0880	00 01DE 7FFF	30K-128	Reserved	Reserved
00 01DE 8000	00 01DE 807F	128	Tracer CFG31	Tracer CFG31
00 01DE 8080	00 01DF FFFF	64K-128	Reserved	Reserved
00 01E0 0000	00 01E3 FFFF	256K	Reserved	Reserved
00 01E4 0000	00 01E7FFFF	256k	TSIP_CFG	TSIP_CFG
00 01E8 0000	00 01E8 3FFF	16K	ARM CorePac_CFG	ARM CorePac_CFG
00 01E8 4000	00 01EB FFFF	240k	Reserved	Reserved
00 01EC 0000	00 01EF FFFF	256K	Reserved	Reserved
00 01F0 0000	00 01F7 FFFF	512K	Reserved	Reserved
00 01F8 0000	00 01F8 FFFF	64K	Reserved	Reserved
00 01F9 0000	00 01F9 FFFF	64K	Reserved	Reserved
00 01FA 0000	00 01FB FFFF	128K	Reserved	Reserved
00 01FC 0000	00 01FD FFFF	128K	Reserved	Reserved
00 01FE 0000	00 01FF FFFF	128K	Reserved	Reserved
00 0200 0000	00 020F FFFF	1M	Network Coprocessor 0(Packet Accelerator, 1-gigabit Ethernet switch subsystem and Security Accelerator)	Network Coprocessor 0(Packet Accelerator, 1-gigabit Ethernet switch subsystem and Security Accelerator)
00 0210 0000	00 0210 FFFF	64K	Reserved	Reserved
00 0211 0000	00 0211 FFFF	64K	Reserved	Reserved
00 0212 0000	00 0213 FFFF	128K	Reserved	Reserved
00 0214 0000	00 0215 FFFF	128K	Reserved	Reserved
00 0216 0000	00 0217 FFFF	128K	Reserved	Reserved

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 0218 0000	00 0218 7FFF	32k	Reserved	Reserved
00 0218 8000	00 0218 FFFF	32k	Reserved	Reserved
00 0219 0000	00 0219 FFFF	64k	Reserved	Reserved
00 021A 0000	00 021A FFFF	64K	Reserved	Reserved
00 021B 0000	00 021B FFFF	64K	Reserved	Reserved
00 021C 0000	00 021C 03FF	1K	Reserved	Reserved
00 021C 0400	00 021C 3FFF	15K	Reserved	Reserved
00 021C 4000	00 021C 43FF	1K	Reserved	Reserved
00 021C 4400	00 021C 5FFF	7K	Reserved	Reserved
00 021C 6000	00 021C 63FF	1K	Reserved	Reserved
00 021C 6400	00 021C 7FFF	7K	Reserved	Reserved
00 021C 8000	00 021C 83FF	1K	Reserved	Reserved
00 021C 8400	00 021C FFFF	31K	Reserved	Reserved
00 021D 0000	00 021D 03FF	1K	Memory protection unit (MPU) 15	Memory protection unit (MPU) 15
00 021D 0400	00 021D 047F	128	Tracer CFG32	Tracer CFG32
00 021D 0100	00 021D 3FFF	15K-128	Reserved	Reserved
00 021D 4000	00 021D 40FF	256	Reserved	Reserved
00 021D 4100	00 021D 7FFF	16K-256	Reserved	Reserved
00 021D 8000	00 021D 80FF	256	Reserved	Reserved
00 021D 8100	00 021D BFFF	16K-256	Reserved	Reserved
00 021D C000	00 021D C0FF	256	Reserved	Reserved
00 021D C100	00 021D EFFF	12K-256	Reserved	Reserved
00 021D F000	00 021D F07F	128	USIM configuration	USIM configuration
00 021D F080	00 021D FFFF	4K-128	Reserved	Reserved
00 021E 0000	00 021E FFFF	64K	Reserved	Reserved
00 021F 0000	00 021F 07FF	2K	Reserved	Reserved
00 021F 0800	00 021F 0FFF	2K	Reserved	Reserved
00 021F 1000	00 021F 17FF	2K	Reserved	Reserved
00 021F 1800	00 021F 3FFF	10K	Reserved	Reserved
00 021F 4000	00 021F 47FF	2K	Reserved	Reserved
00 021F 4800	00 021F 7FFF	14K	Reserved	Reserved
00 021F 8000	00 021F 87FF	2K	Reserved	Reserved
00 021F 8800	00 021F BFFF	14K	Reserved	Reserved
00 021F C000	00 021F C7FF	2K	Reserved	Reserved
00 021F C800	00 021F FFFF	14K	Reserved	Reserved
00 0220 0000	00 0220 007F	128	Reserved	Reserved
00 0220 0080	00 0220 FFFF	64K-128	Reserved	Reserved
00 0221 0000	00 0221 007F	128	Reserved	Reserved
00 0221 0080	00 0221 FFFF	64K-128	Reserved	Reserved
00 0222 0000	00 0222 007F	128	Reserved	Reserved
00 0222 0080	00 0222 FFFF	64K-128	Reserved	Reserved
00 0223 0000	00 0223 007F	128	Reserved	Reserved
00 0223 0080	00 0223 FFFF	64K-128	Reserved	Reserved
00 0224 0000	00 0224 007F	128	Reserved	Reserved
00 0224 0080	00 0224 FFFF	64K-128	Reserved	Reserved
00 0225 0000	00 0225 007F	128	Reserved	Reserved
00 0225 0080	00 0225 FFFF	64K-128	Reserved	Reserved
00 0226 0000	00 0226 007F	128	Reserved	Reserved
00 0226 0080	00 0226 FFFF	64K-128	Reserved	Reserved
00 0227 0000	00 0227 007F	128	Reserved	Reserved

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 0227 0080	00 0227 FFFF	64K-128	Reserved	Reserved
00 0228 0000	00 0228 007F	128	Timer 8	Timer 8
00 0228 0080	00 0228 FFFF	64K-128	Reserved	Reserved
00 0229 0000	00 0229 007F	128	Timer 9	Timer 9
00 0229 0080	00 0229 FFFF	64K-128	Reserved	Reserved
00 022A 0000	00 022A 007F	128	Timer 10	Timer 10
00 022A 0080	00 022A FFFF	64K-128	Reserved	Reserved
00 022B 0000	00 022B 007F	128	Timer 11	Timer 11
00 022B 0080	00 022B FFFF	64K-128	Reserved	Reserved
00 022C 0000	00 022C 007F	128	Timer 12	Timer 12
00 022C 0080	00 022C FFFF	64K-128	Reserved	Reserved
00 022D 0000	00 022D 007F	128	Timer 13	Timer 13
00 022D 0080	00 022D FFFF	64K-128	Reserved	Reserved
00 022E 0000	00 022E 007F	128	Timer 14	Timer 14
00 022E 0080	00 022E FFFF	64K-128	Reserved	Reserved
00 022F 0000	00 022F 007F	128	Timer 15	Timer 15
00 022F 0080	00 022F 00FF	128	Timer 16	Timer 16
00 022F 0100	00 022F 017F	128	Timer 17	Timer 17
00 022F 0180	00 022F 01FF	128	Timer 18	Timer 18
00 022F 0200	00 022F 027F	128	Timer 19	Timer 19
00 0230 0000	00 0230 FFFF	64K	Reserved	Reserved
00 0231 0000	00 0231 01FF	512	PLL Controller	PLL Controller
00 0231 0200	00 0231 9FFF	40K-512	Reserved	Reserved
00 0231 A000	00 0231 BFFF	8K	HyperLink0 SerDes Config	HyperLink0 SerDes Config
00 0231 C000	00 0231 DFFF	8K	Reserved	Reserved
00 0231 E000	00 0231 FFFF	8K	10GbE SerDes Config	10GbE SerDes Config
00 0232 0000	00 0232 3FFF	16K	PCIe0 SerDes Config	PCIe0 SerDes Config
00 0232 4000	00 0232 5FFF	8K	SGMII 1 SerDes Config	SGMII 1 SerDes Config
00 0232 6000	00 0232 7FFF	8K	PCIe1SerDes Config	PCIe1SerDes Config
00 0232 8000	00 0232 8FFF	4K	Reserved	Reserved
00 0232 9000	00 0232 9FFF	4K	DDRA PHY Config	DDRA PHY Config
00 0232 A000	00 0232 BFFF	8K	SGMII 0 SerDes Config	SGMII 0 SerDes Config
00 0232 C000	00 0232 CFFF	4K	Reserved	Reserved
00 0232 D000	00 0232 DFFF	4K	Reserved	Reserved
00 0232 E000	00 0232 EFFF	8K	Reserved	Reserved
00 0233 0000	00 0233 03FF	1K	SmartReflex0	SmartReflex0
00 0233 0400	00 0233 07FF	1K	Reserved	Reserved
00 0233 0400	00 0233 FFFF	62K	Reserved	Reserved
00 0234 0000	00 0234 00FF	256	Reserved	Reserved
00 0234 0100	00 0234 3FFF	16K	Reserved	Reserved
00 0234 4000	00 0234 40FF	256	Reserved	Reserved
00 0234 4100	00 0234 7FFF	16K	Reserved	Reserved
00 0234 8000	00 0234 80FF	256	Reserved	Reserved
00 0234 8100	00 0234 BFFF	16K	Reserved	Reserved
00 0234 C000	00 0234 C0FF	256	Reserved	Reserved
00 0234 C100	00 0234 FFFF	16K	Reserved	Reserved
00 0235 0000	00 0235 0FFF	4K	Power sleep controller (PSC)	Power sleep controller (PSC)
00 0235 1000	00 0235 FFFF	64K-4K	Reserved	Reserved
00 0236 0000	00 0236 03FF	1K	Memory protection unit (MPU) 0	Memory protection unit (MPU) 0
00 0236 0400	00 0236 7FFF	31K	Reserved	Reserved

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 0236 8000	00 0236 83FF	1K	Memory protection unit (MPU) 1	Memory protection unit (MPU) 1
00 0236 8400	00 0236 FFFF	31K	Reserved	Reserved
00 0237 0000	00 0237 03FF	1K	Memory protection unit (MPU) 2	Memory protection unit (MPU) 2
00 0237 0400	00 0237 7FFF	31K	Reserved	Reserved
00 0237 8000	00 0237 83FF	1K	Reserved	Reserved
00 0237 8400	00 0237 FFFF	31K	Reserved	Reserved
00 0238 0000	00 0238 03FF	1K	Reserved	Reserved
00 0238 8000	00 0238 83FF	1K	Memory protection unit (MPU) 5	Memory protection unit (MPU) 5
00 0238 8400	00 0238 87FF	1K	Reserved	Reserved
00 0238 8800	00 0238 8BFF	1K	Memory protection unit (MPU) 7	Memory protection unit (MPU) 7
00 0238 8C00	00 0238 8FFF	1K	Memory protection unit (MPU) 8	Memory protection unit (MPU) 8
00 0238 9000	00 0238 93FF	1K	Memory protection unit (MPU) 9	Memory protection unit (MPU) 9
00 0238 9400	00 0238 97FF	1K	Memory protection unit (MPU) 10	Memory protection unit (MPU) 10
00 0238 9800	00 0238 9BFF	1K	Memory protection unit (MPU) 11	Memory protection unit (MPU) 11
00 0238 9C00	00 0238 9FFF	1K	Memory protection unit (MPU) 12	Memory protection unit (MPU) 12
00 0238 A000	00 0238 A3FF	1K	Memory protection unit (MPU) 13	Memory protection unit (MPU) 13
00 0238 A400	00 0238 A7FF	1K	Memory protection unit (MPU) 14	Memory protection unit (MPU) 14
00 0238 A800	00 023F FFFF	471K	Reserved	Reserved
00 0240 0000	00 0243 FFFF	256K	Reserved	Reserved
00 0244 0000	00 0244 3FFF	16K	Reserved	Reserved
00 0244 4000	00 0244 FFFF	48K	Reserved	Reserved
00 0245 0000	00 0245 3FFF	16K	Reserved	Reserved
00 0245 4000	00 0245 FFFF	48K	Reserved	Reserved
00 0246 0000	00 0246 3FFF	16K	Reserved	Reserved
00 0246 4000	00 0246 FFFF	48K	Reserved	Reserved
00 0247 0000	00 0247 3FFF	16K	Reserved	Reserved
00 0247 4000	00 0247 FFFF	48K	Reserved	Reserved
00 0248 0000	00 0248 3FFF	16K	Reserved	Reserved
00 0248 4000	00 0248 FFFF	48K	Reserved	Reserved
00 0249 0000	00 0249 3FFF	16K	Reserved	Reserved
00 0249 4000	00 0249 FFFF	48K	Reserved	Reserved
00 024A 0000	00 024A 3FFF	16K	Reserved	Reserved
00 024A 4000	00 024A FFFF	48K	Reserved	Reserved
00 024B 0000	00 024B 3FFF	16K	Reserved	Reserved
00 024B 4000	00 024B FFFF	48K	Reserved	Reserved
00 024C 0000	00 024C 01FF	512	Reserved	Reserved
00 024C 0200	00 024C 03FF	1K-512	Reserved	Reserved
00 024C 0400	00 024C 07FF	1K	Reserved	Reserved
00 024C 0800	00 024C FFFF	62K	Reserved	Reserved
00 024D 0000	00 024F FFFF	192K	Reserved	Reserved
00 0250 0000	00 0250 007F	128	Reserved	Reserved
00 0250 0080	00 0250 7FFF	32K-128	Reserved	Reserved
00 0250 8000	00 0250 FFFF	32K	Reserved	Reserved
00 0251 0000	00 0251 FFFF	64K	Reserved	Reserved
00 0252 0000	00 0252 03FF	1K	Reserved	Reserved
00 0252 0400	00 0252 FFFF	64K-1K	Reserved	Reserved
00 0253 0000	00 0253 007F	128	I^2C0	I^2C0
00 0253 0080	00 0253 03FF	1K-128	Reserved	Reserved
00 0253 0400	00 0253 047F	128	I^2C1	I^2C1
00 0253 0480	00 0253 07FF	1K-128	Reserved	Reserved

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 0253 0800	00 0253 087F	128	I^2C2	I^2C2
00 0253 0880	00 0253 0BFF	1K-128	Reserved	Reserved
00 0253 0C00	00 0253 0C3F	64	UART0	UART0
00 0253 0C40	00 0253 FFFF	1K-64	Reserved	Reserved
00 0253 1000	00 0253 103F	64	UART1	UART1
00 0253 1040	00 0253 FFFF	60K-64	Reserved	Reserved
00 0254 0000	00 0255 FFFF	128K	Reserved	Reserved
00 0256 0080	00 0257 FFFF	128K	ARM CorePac INTC	ARM CorePac INTC
00 0258 0000	00 025F FFFF	512K	Reserved	Reserved
00 0260 0000	00 0260 1FFF	8K	Secondary interrupt controller (CIC) 0	Secondary interrupt controller (CIC) 0
00 0260 2000	00 0260 3FFF	8K	Reserved	Reserved
00 0260 4000	00 0260 5FFF	8K	Reserved	Reserved
00 0260 6000	00 0260 7FFF	8K	Reserved	Reserved
00 0260 8000	00 0260 9FFF	8K	Secondary interrupt controller (CIC) 2	Secondary interrupt controller (CIC) 2
00 0260 A000	00 0260 BEFF	8K-256	Reserved	Reserved
00 0260 BF00	00 0260 BFFF	256	GPIO Config	GPIO Config
00 0260 C000	00 0261 BFFF	64K	Reserved	Reserved
00 0261 C000	00 0261 FFFF	16K	Reserved	Reserved
00 0262 0000	00 0262 0FFF	4K	BOOTCFG chip-level registers	BOOTCFG chip-level registers
00 0262 1000	00 0262 FFFF	60K	Reserved	Reserved
00 0263 0000	00 0263 FFFF	64K	USB 0 PHY CFG	USB 0 PHY CFG
00 0264 0000	00 0264 07FF	2K	Semaphore Config	Semaphore Config
00 0264 0800	00 0264 FFFF	62K	Reserved	Reserved
00 0265 0000	00 0267 FFFF	192K	Reserved	Reserved
00 0268 0000	00 026F FFFF	512K	USB 0 MMR CFG	USB 0 MMR CFG
00 0270 0000	00 0270 7FFF	32K	EDMA channel controller (TPCC) 0	EDMA channel controller (TPCC) 0
00 0270 8000	00 0270 FFFF	32K	EDMA channel controller (TPCC) 4	EDMA channel controller (TPCC) 4
00 0271 0000	00 0271 FFFF	64K	Reserved	Reserved
00 0272 0000	00 0272 7FFF	32K	EDMA channel controller (TPCC) 1	EDMA channel controller (TPCC) 1
00 0272 8000	00 0272 FFFF	32K	EDMA channel controller (TPCC) 3	EDMA channel controller (TPCC) 3
00 0273 0000	00 0273 FFFF	64K	Reserved	Reserved
00 0274 0000	00 0274 7FFF	32K	EDMA channel controller (TPCC) 2	EDMA channel controller (TPCC) 2
00 0274 8000	00 0275 FFFF	96K	Reserved	Reserved
00 0276 0000	00 0276 03FF	1K	EDMA TPCC0 transfer controller (TPTC) 0	EDMA TPCC0 transfer controller (TPTC) 0
00 0276 0400	00 0276 7FFF	31K	Reserved	Reserved
00 0276 8000	00 0276 83FF	1K	EDMA TPCC0 transfer controller (TPTC) 1	EDMA TPCC0 transfer controller (TPTC) 1
00 0276 8400	00 0276 FFFF	31K	Reserved	Reserved
00 0277 0000	00 0277 03FF	1K	EDMA TPCC1 transfer controller (TPTC) 0	EDMA TPCC1 transfer controller (TPTC) 0
00 0277 0400	00 0277 7FFF	31K	Reserved	Reserved
00 0277 8000	00 0277 83FF	1K	EDMA TPCC1 transfer controller (TPTC) 1	EDMA TPCC1 transfer controller (TPTC) 1
00 0278 0400	00 0277 FFFF	31K	Reserved	Reserved
00 0278 0000	00 0278 03FF	1K	EDMA TPCC1 transfer controller (TPTC) 2	EDMA TPCC1 transfer controller (TPTC) 2
00 0278 0400	00 0278 7FFF	31K	Reserved	Reserved
00 0278 8000	00 0278 83FF	1K	EDMA TPCC1 transfer controller (TPTC) 3	EDMA TPCC1 transfer controller (TPTC) 3
00 0278 8400	00 0278 FFFF	31K	Reserved	Reserved
00 0279 0000	00 0279 03FF	1K	EDMA TPCC2 transfer controller (TPTC) 0	EDMA TPCC2 transfer controller (TPTC) 0
00 0279 0400	00 0279 7FFF	31K	Reserved	Reserved
00 0279 8000	00 0279 83FF	1K	EDMA TPCC2 transfer controller (TPTC) 1	EDMA TPCC2 transfer controller (TPTC) 1
00 0279 8400	00 0279 FFFF	31K	Reserved	Reserved
00 027A 0000	00 027A 03FF	1K	EDMA TPCC2 transfer controller (TPTC) 2	EDMA TPCC2 transfer controller (TPTC) 2

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 027A 0400	00 027A 7FFF	31K	Reserved	Reserved
00 027A 8000	00 027A 83FF	1K	EDMA TPCC2 transfer controller (TPTC) 3	EDMA TPCC2 transfer controller (TPTC) 3
00 027A 8400	00 027A FFFF	31K	Reserved	Reserved
00 027B 0000	00 027B 03FF	1K	EDMA TPCC3 transfer controller (TPTC) 0	EDMA TPCC3 transfer controller (TPTC) 0
00 027B 0400	00 027B 7FFF	31K	Reserved	Reserved
00 027B 8000	00 027B 83FF	1K	EDMA TPCC3 transfer controller (TPTC) 1	EDMA TPCC3 transfer controller (TPTC) 1
00 027B 8400	00 027B 87FF	1K	EDMA TPCC4 transfer controller (TPTC) 0	EDMA TPCC4 transfer controller (TPTC) 0
00 027B 8800	00 027B 8BFF	1K	EEDMA TPCC4 transfer controller (TPTC) 1	EEDMA TPCC4 transfer controller (TPTC) 1
00 027B 8C00	00 027B FFFF	29K	Reserved	Reserved
00 027C 0000	00 027C 03FF	1K	Reserved	Reserved
00 027C 0400	00 027C FFFF	63K	Reserved	Reserved
00 027D 0000	00 027D 3FFF	16K	TI embedded trace buffer (TETB) - CorePac0	TI embedded trace buffer (TETB) - CorePac0
00 027D 4000	00 027D 7FFF	16K	TBR_ARM CorePac - Trace buffer - ARM CorePac	TBR_ARM CorePac - Trace buffer - ARM CorePac
00 027D 8000	00 027D FFFF	32K	Reserved	Reserved
00 027E 0000	00 027E 3FFF	16K	Reserved	Reserved
00 027E 4000	00 027E FFFF	48K	Reserved	Reserved
00 027F 0000	00 027F 3FFF	16K	Reserved	Reserved
00 027F 4000	00 027F FFFF	48K	Reserved	Reserved
00 0280 0000	00 0280 3FFF	16K	Reserved	Reserved
00 0280 4000	00 0280 FFFF	48K	Reserved	Reserved
00 0281 0000	00 0281 3FFF	16K	Reserved	Reserved
00 0281 4000	00 0281 FFFF	48K	Reserved	Reserved
00 0282 0000	00 0282 3FFF	16K	Reserved	Reserved
00 0282 4000	00 0282 FFFF	48K	Reserved	Reserved
00 0283 0000	00 0283 3FFF	16K	Reserved	Reserved
00 0283 4000	00 0283 FFFF	48K	Reserved	Reserved
00 0284 0000	00 0284 3FFF	16K	Reserved	Reserved
00 0284 4000	00 0284 FFFF	48K	Reserved	Reserved
00 0285 0000	00 0285 7FFF	32K	TBR_SYS- Trace buffer - System	TBR_SYS- Trace buffer - System
00 0285 8000	00 0285 FFFF	32K	Reserved	Reserved
00 0286 0000	00 028F FFFF	640K	Reserved	Reserved
00 0290 0000	00 0293 FFFF	256K	Reserved	Reserved
00 0294 0000	00 029F FFFF	768K	Reserved	Reserved
00 02A0 0000	00 02AF FFFF	1M	Navigator configuration	Navigator configuration
00 02B0 0000	00 02BF FFFF	1M	Navigator linking RAM	Navigator linking RAM
00 02C0 0000	00 02C0 FFFF	64K	Reserved	Reserved
00 02C1 0000	00 02C1 FFFF	64K	Reserved	Reserved
00 02C2 0000	00 02C3 FFFF	128K	Reserved	Reserved
00 02C4 0000	00 02C5 FFFF	128K	Reserved	Reserved
00 02C6 0000	00 02C7 FFFF	128K	Reserved	Reserved
00 02C8 0000	00 02C8 FFFF	64K	Reserved	Reserved
00 02C9 0000	00 02C9 FFFF	64K	Reserved	Reserved
00 02CA 0000	00 02CB FFFF	128K	Reserved	Reserved
00 02CC 0000	00 02CD FFFF	128K	Reserved	Reserved
00 02CE 0000	00 02EF FFFF	15M-896K	Reserved	Reserved
00 02F0 0000	00 02FF FFFF	1M	10GbE Config	10GbE Config
00 0300 0000	00 030F FFFF	1M	DBG Config	DBG Config
00 0310 0000	00 07FF FFFF	79M	Reserved	Reserved
00 0800 0000	00 0801 FFFF	128K	Extended memory controller (XMC) configuration	Extended memory controller (XMC) configuration

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 0802 0000	00 0BBF FFFF	60M-128K	Reserved	Reserved
00 0BC0 0000	00 0BCF FFFF	1M	Multicore shared memory controller (MSMC) config	Multicore shared memory controller (MSMC) config
00 0BD0 0000	00 0BFF FFFF	3M	Reserved	Reserved
00 0C00 0000	00 0C1F FFFF	2M	Multicore shared memory (MSM)	Multicore shared memory (MSM)
00 0C20 0000	00 0C5F FFFF	4M	Reserved	Reserved
00 0C60 0000	00 0FFF FFFF	58M	Reserved	Reserved
00 1000 0000	00 107F FFFF	8M	Reserved	Reserved
00 1080 0000	00 108F FFFF	1M	Reserved	Reserved
00 1090 0000	00 10DF FFFF	5M	Reserved	Reserved
00 10E0 0000	00 10E0 7FFF	32K	Reserved	Reserved
00 10E0 8000	00 10EF FFFF	1M-32K	Reserved	Reserved
00 10F0 0000	00 10F0 7FFF	32K	Reserved	Reserved
00 10F0 8000	00 117F FFFF	9M-32K	Reserved	Reserved
00 1180 0000	00 118F FFFF	1M	Reserved	Reserved
00 1190 0000	00 11DF FFFF	5M	Reserved	Reserved
00 11E0 0000	00 11E0 7FFF	32K	Reserved	Reserved
00 11E0 8000	00 11EF FFFF	1M-32K	Reserved	Reserved
00 11F0 0000	00 11F0 7FFF	32K	Reserved	Reserved
00 11F0 8000	00 127F FFFF	9M-32K	Reserved	Reserved
00 1280 0000	00 128F FFFF	1M	Reserved	Reserved
00 1290 0000	00 12DF FFFF	5M	Reserved	Reserved
00 12E0 0000	00 12E0 7FFF	32K	Reserved	Reserved
00 12E0 8000	00 12EF FFFF	1M-32K	Reserved	Reserved
00 12F0 0000	00 12F0 7FFF	32K	Reserved	Reserved
00 12F0 8000	00 137F FFFF	9M-32K	Reserved	Reserved
00 1380 0000	00 1388 FFFF	1M	Reserved	Reserved
00 1390 0000	00 13DF FFFF	5M	Reserved	Reserved
00 13E0 0000	00 13E0 7FFF	32K	Reserved	Reserved
00 13E0 8000	00 13EF FFFF	1M-32K	Reserved	Reserved
00 13F0 0000	00 13F0 7FFF	32K	Reserved	Reserved
00 13F0 8000	00 147F FFFF	9M-32K	Reserved	Reserved
00 1480 0000	00 148F FFFF	1M	Reserved	Reserved
00 1490 0000	00 14DF FFFF	5M	Reserved	Reserved
00 14E0 0000	00 14E0 7FFF	32K	Reserved	Reserved
00 14E0 8000	00 14EF FFFF	1M-32K	Reserved	Reserved
00 14F0 0000	00 14F0 7FFF	32K	Reserved	Reserved
00 14F0 8000	00 157F FFFF	9M-32K	Reserved	Reserved
00 1580 0000	00 158F FFFF	1M	Reserved	Reserved
00 1590 0000	00 15DF FFFF	5M	Reserved	Reserved
00 15E0 0000	00 15E0 7FFF	32K	Reserved	Reserved
00 15E0 8000	00 15EF FFFF	1M-32K	Reserved	Reserved
00 15F0 0000	00 15F0 7FFF	32K	Reserved	Reserved
00 15F0 8000	00 167F FFFF	9M-32K	Reserved	Reserved
00 1680 0000	00 168F FFFF	1M	Reserved	Reserved
00 1690 0000	00 16DF FFFF	5M	Reserved	Reserved
00 16E0 0000	00 16E0 7FFF	32K	Reserved	Reserved
00 16E0 8000	00 16EF FFFF	1M-32K	Reserved	Reserved
00 16F0 0000	00 16F0 7FFF	32K	Reserved	Reserved
00 16F0 8000	00 177F FFFF	9M-32K	Reserved	Reserved
00 1780 0000	00 178F FFFF	1M	Reserved	Reserved

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 1790 0000	00 17DF FFFF	5M	Reserved	Reserved
00 17E0 0000	00 17E0 7FFF	32K	Reserved	Reserved
00 17E0 8000	00 17EF FFFF	1M-32K	Reserved	Reserved
00 17F0 0000	00 17F0 7FFF	32K	Reserved	Reserved
00 17F0 8000	00 1FFF FFFF	129M-32K	Reserved	Reserved
00 2000 0000	00 200F FFFF	1M	System trace manager (STM) configuration	System trace manager (STM) configuration
00 2010 0000	00 201F FFFF	1M	Reserved	Reserved
00 2020 0000	00 205F FFFF	4M	Reserved	Reserved
00 2060 0000	00 206F FFFF	1M	Reserved	Reserved
00 2070 0000	00 2077 FFFF	512K	Reserved	Reserved
00 2078 0000	00 2078 FFFF	64K	Reserved	Reserved
00 2079 0000	00 207F FFFF	448K	Reserved	Reserved
00 2080 0000	00 208F FFFF	1M	Reserved	Reserved
00 2090 0000	00 209F FFFF	1M	Reserved	Reserved
00 20A0 0000	00 20A3 FFFF	256K	Reserved	Reserved
00 20A4 0000	00 20A4 FFFF	64K	Reserved	Reserved
00 20A5 0000	00 20AF FFFF	704K	Reserved	Reserved
00 20B0 0000	00 20B3 FFFF	256K	Boot ROM	Boot ROM
00 20B4 0000	00 20BE FFFF	704K	Reserved	Reserved
00 20BF 0000	00 20BF 01FF	64K	Reserved	Reserved
00 20C0 0000	00 20FF FFFF	4M	Reserved	Reserved
00 2100 0000	00 2100 03FF	1K	Reserved	Reserved
00 2100 0400	00 2100 05FF	512	SPI0	SPI0
00 2100 0600	00 2100 07FF	512	SPI1	SPI1
00 2100 0800	00 2100 09FF	512	SPI2	SPI2
00 2100 0A00	00 2100 0AFF	256	EMIF Config	EMIF Config
00 2100 0B00	00 2100 FFFF	62K-768	Reserved	Reserved
00 2101 0000	00 2101 01FF	512	DDR3 EMIF Config	DDR3 EMIF Config
00 2101 0200	00 2101 07FF	2K-512	Reserved	Reserved
00 2101 0800	00 2101 09FF	512	Reserved	Reserved
00 2101 0A00	00 2101 0FFF	2K-512	Reserved	Reserved
00 2101 1000	00 2101 FFFF	60K	Reserved	Reserved
00 2102 0000	00 2102 7FFF	32K	PCIe 1 config	PCIe 1 config
00 2102 8000	00 2103 FFFF	96K	Reserved	Reserved
00 2104 0000	00 217F FFFF	4M-256K	Reserved	Reserved
00 2140 0000	00 2140 00FF	256	HyperLink0 config	HyperLink0 config
00 2140 0100	00 2140 01FF	256	Reserved	Reserved
00 2140 0400	00 217F FFFF	4M-512	Reserved	Reserved
00 2180 0000	00 2180 7FFF	32K	PCIe 0 config	PCIe 0 config
00 2180 8000	00 21BF FFFF	4M-32K	Reserved	Reserved
00 21C0 0000	00 21FF FFFF	4M	Reserved	Reserved
00 2200 0000	00 229F FFFF	10M	Reserved	Reserved
00 22A0 0000	00 22A0 FFFF	64K	Reserved	Reserved
00 22A1 0000	00 22AF FFFF	1M-64K	Reserved	Reserved
00 22B0 0000	00 22B0 FFFF	64K	Reserved	Reserved
00 22B1 0000	00 22BF FFFF	1M-64K	Reserved	Reserved
00 22C0 0000	00 22C0 FFFF	64K	Reserved	Reserved
00 22C1 0000	00 22CF FFFF	1M-64K	Reserved	Reserved
00 22D0 0000	00 22D0 FFFF	64K	Reserved	Reserved
00 22D1 0000	00 22DF FFFF	1M-64K	Reserved	Reserved

Table 6-1. Device Memory Map Summary AM5K2E0x (continued)

Physical 40-bit Address		Bytes	ARM View	SOC View
Start	End			
00 22E0 0000	00 22E0 FFFF	64K	Reserved	Reserved
00 22E1 0000	00 22EF FFFF	1M-64K	Reserved	Reserved
00 22F0 0000	00 22F0 FFFF	64K	Reserved	Reserved
00 22F1 0000	00 22FF FFFF	1M-64K	Reserved	Reserved
00 2300 0000	00 2300 FFFF	64K	Reserved	Reserved
00 2301 0000	00 230F FFFF	1M-64K	Reserved	Reserved
00 2310 0000	00 2310 FFFF	64K	Reserved	Reserved
00 2311 0000	00 231F FFFF	1M-64K	Reserved	Reserved
00 2320 0000	00 2324 FFFF	384K	Reserved	Reserved
00 2325 0000	00 239F FFFF	8M-384K	Reserved	Reserved
00 23A0 0000	00 23BF FFFF	2M	Navigator	Navigator
00 23C0 0000	00 23FF FFFF	4M	Reserved	Reserved
00 2400 0000	00 24FF FFFF	16M	NETCP15 config	NETCP15 config
00 2500 0000	00 2507 FFFF	512K	USB 1 MMR config	USB 1 MMR config
00 2508 0000	00 2508 FFFF	64K	USB 1 PHY config	USB 1 PHY config
00 2509 0000	00 27FF FFFF	48M-576K	Reserved	Reserved
00 2800 0000	00 2FFF FFFF	128M	Reserved	Reserved
00 3000 0000	00 33FF FFFF	64M	EMIF16 CE0	EMIF16 CE0
00 3400 0000	00 37FF FFFF	64M	EMIF16 CE1	EMIF16 CE1
00 3800 0000	00 3BFF FFFF	64M	EMIF16 CE2	EMIF16 CE2
00 3C00 0000	00 3FFF FFFF	64M	EMIF16 CE3	EMIF16 CE3
00 4000 0000	00 4FFF FFFF	256M	HyperLink0 data	HyperLink0 data
00 5000 0000	00 5FFF FFFF	256M	PCIe 0 data	PCIe 0 data
00 6000 0000	00 6FFF FFFF	256M	PCIe 1data	PCIe 1data
00 7000 0000	00 FFFF FFFF	2304M	Reserved	Reserved
01 0000 0000	01 20FF FFFF	528M	Reserved	Reserved
01 2100 0000	01 2100 01FF	512	Reserved	DDR3 EMIF configuration
01 2100 0200	07 FFFF FFFF	32G-512	Reserved	Reserved
08 0000 0000	09 FFFF FFFF	8G	DDR3 data	DDR3 data
0A 0000 0000	FF FFFF FFFF	984G	Reserved	Reserved

6.2 Memory Protection Unit (MPU) for AM5K2E0x

CFG (configuration) space of all slave devices on the TeraNet is protected by the MPU. The AM5K2E0x contains sixteen MPUs of which thirteen MPUs are used:

- MPU0 is used to protect main CORE/3 CFG TeraNet_3P_B (SCR_3P (B)).
- MPU1/2/5 are used for QM_SS (one for VBUSM port and one each for the two configuration VBUSP port).
- MPU3/4/6 are not used.
- MPU7 is used for PCIe1.
- MPU8 is used for peripherals connected to TeraNet_6P_A (SCR_6P (A)).
- MPU9 is used for interrupt controllers connected to TeraNet_3P (SCR_3P).
- MPU10 is used for semaphore.
- MPU11 is used to protect TeraNet_6P_B (SCR_6P (B)) CPU/6 CFG TeraNet.
- MPU12/13/14 are used for SPI0/1/2.
- MPU15 is used for USB1.

This section contains MPU register map and details of device-specific MPU registers only. For MPU features and details of generic MPU registers, see the *KeyStone Architecture Memory Protection Unit (MPU) User's Guide* (SPRUGW5).

The following tables show the configuration of each MPU and the memory regions protected by each MPU.

Table 6-2. MPU0-MPU5 Default Configuration

SETTING	MPU0 MAIN SCR_3P (B)	MPU1 QM_SS DATA PORT	MPU2 QM_SS CFG1 PORT	MPU3	MPU4	MPU5 QM_SS CFG2 PORT
Default permission	Assume allowed	Assume allowed	Assume allowed	Reserved	Reserved	Assume allowed
Number of allowed IDs supported	16	16	16			16
Number of programmable ranges supported	16	16	16			16
Compare width	1KB granularity	1KB granularity	1KB granularity			1KB granularity

Table 6-3. MPU6-MPU11 Default Configuration

SETTING	MPU6	MPU7 PCIe1	MPU8 EMIF16	MPU9 CIC	MPU10 SM	MPU11 SCR_6P (B)
Default permission	Reserved	Assume allowed	Assume allowed	Assume allowed	Assume allowed	Assume allowed
Number of allowed IDs supported		16	16	16	16	16
Number of programmable ranges supported		16	8	4	2	16
Compare width		1KB granularity	1KB granularity	1KB granularity	1KB granularity	1KB granularity

Table 6-4. MPU12-MPU15 Default Configuration

SETTING	MPU12 SPI0	MPU13 SPI1	MPU14 SPI2	MPU15 USB1
Default permission	Assume allowed	Assume allowed	Assume allowed	Assume allowed
Number of allowed IDs supported	16	16	16	16
Number of programmable ranges supported	2	2	2	8
Compare width	1KB granularity	1KB granularity	1KB granularity	1KB granularity

Table 6-5. MPU Memory Regions

	MEMORY PROTECTION	START ADDRESS	END ADDRESS
MPU0	Main CFG SCR	0x01D0_0000	0X01E7_FFFF
MPU1	QM_SS DATA PORT	0x23A0_0000	0x23BF_FFFF
MPU2	QM_SS CFG1 PORT	0x02A0_0000	0x02AF_FFFF
MPU3	Reserved	N/A	N/A
MPU4	Reserved	N/A	N/A
MPU5	QM_SS CFG2 PORT	0x02A0_4000	0x02BF_FFFF
MPU6	Reserved	N/A	N/A
MPU7	PCIe1	0x2101_0000	0xFFFF_FFFF
MPU8	SPIROM/EMIF16	0x20B0_0000	0x3FFF_FFFF
MPU9	CIC/AINTC	0x0264_0000	0x0264_07FF
MPU10	Semaphore	0x0260_0000	0x0260_9FFF
MPU11	SCR_6 and CPU/6 CFG SCR	0x0220_0000	0x03FF_FFFF
MPU12	SPI0	0x2100_0400	0x2100_07FF
MPU13	SPI1	0x2100_0400	0x2100_07FF
MPU14	SPI2	0x2100_0800	0x2100_0AFF

Table 6-5. MPU Memory Regions (continued)

	MEMORY PROTECTION	START ADDRESS	END ADDRESS
MPU15	USB1	0x2400_0000	0x2508_FFFF

Table 6-6 shows the unique Master ID assigned to each CorePac and peripherals on the device.

Table 6-6. Master ID Settings

Master ID	AM5K2E0x
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	ARM CorePac 0
9	ARM CorePac1
10	ARM CorePac 2
11	ARM CorePac 3
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	EDMA0_TC0 read
26	EDMA0_TC0 write
27	EDMA0_TC1 read
28	Hyperlink0
29	USB1
30	Reserved
31	PCIe0
32	EDMA0_TC1 write
33	EDMA1_TC0 read
34	EDMA1_TC0 write
35	EDMA1_TC1 read
36	EDMA1_TC1write
37	EDMA1_TC2 read
38	EDMA1_TC2 write
39	EDMA1_TC3 read
40	EDMA1_TC3 write
41	EDMA2_TC0 read

Table 6-6. Master ID Settings (continued)

Master ID	AM5K2E0x
42	EDMA2_TC0 write
43	EDMA2_TC1 read
44	EDMA2_TC1 write
45	EDMA2_TC2 read
46	EDMA2_TC2 write
47	EDMA2_TC3 read
48	EDMA2_TC3 write
49	EDMA3_TC0 read
50	EDMA3_TC0 write
51	EDMA3_TC1 read
52	Reserved
53	EDMA3_TC1 write
54-55	Reserved
56	USB0
57	Reserved
58	Reserved
59	Reserved
60	Reserved
61	Reserved
62	EDMA3CC0
63	EDMA3CC1
64	EDMA3CC2
65	Reserved
66	Reserved
67	Reserved
68-71	Queue Manager
72-75	NETCP_GLOBAL1
76-79	Reserved
80	TSIP
81	Reserved
82	Reserved
83	Reserved
84-87	10GbE
88-91	Reserved
92-95	Reserved
96-99	Packet DMA MST1
100-101	Reserved
102	PCIe1
103	Reserved
104	Reserved
105	Reserved
106	Reserved
107	DBG_DAP
108-111	Reserved
112-119	NETCP_LOCAL
120-139	Reserved
140	Reserved

Table 6-6. Master ID Settings (continued)

Master ID	AM5K2E0x
141	Reserved
142	Reserved
143	Reserved
144	Reserved
145	Reserved
146	Reserved
147	Reserved
148	CPT_MSMC0
149	CPT_MSMC1
150	CPT_MSMC2
151	CPT_MSMC3
152	CPT_DDR3
153	CPT_SM
154	CPT_QM_CFG1
155	CPT_QM_M
156	CPT_CFG
157	Reserved
158	Reserved
159	Reserved
160	CPT_QM_CFG2
161	CPT_PCIe1
162	Reserved
163	Reserved
164	CPT_EDMA3CC0_4
165	CPT_EDMA3CC1_2_3
166	CPT_CIC
167	CPT_SPI_ROM_EMIP16
168	Reserved
169	EDMA4_TC0 read
170	EDMA4_TC0 write
171	EDMA4_TC1 read
172	EDMA4_TC1 write
173	EDMA4_CC_TR
174	CPT_MSMC7
175	CPT_MSMC6
176	CPT_MSMC5
177	CPT_MSMC4
178	CPT_NETCP_CFG_MST
179	Reserved
180-183	NETCP_GLOBAL0
184-255	Reserved

Table 6-7 shows the privilege ID of each mastering peripheral. The table also shows the privilege level (supervisor vs. user), security level (secure vs. non-secure), and access type (instruction read vs. data/DMA read or write) of each master on the device. In some cases, a particular setting depends on software being executed at the time of the access or the configuration of the master peripheral.

Table 6-7. Privilege ID Settings

PRIVILEGE ID	MASTER	PRIVILEGE LEVEL	ACCESS TYPE
0	Reserved	N/A	N/A
1	Reserved	N/A	N/A
2	Reserved	N/A	N/A
3	Reserved	N/A	N/A
4	Reserved	N/A	N/A
5	Reserved	N/A	N/A
6	Reserved	N/A	N/A
7	Reserved	N/A	N/A
8	ARM CorePac	User/Supervisor (S/W dependent)	Instruction/Data
9	All packet DMA masters (Both NetCP, Both QM_CDMA) Both USB	User	Data
10	QM_SECOND	User	Data
11	PCIe0	User/Supervisor	Data
12	DAP	User/Supervisor (Emulation S/W dependent)	Data
13	PCIe1	User/Supervisor	Data
14	Hyperlink	User/Supervisor	Data
15	TSIP	User	Data

6.2.1 MPU Registers

This section includes the offsets for MPU registers and definitions for device-specific MPU registers. For Number of Programmable Ranges supported (PROGx_MPSA, PROGxMPEA) refer to the following tables.

6.2.1.1 MPU Register Map

Table 6-8. MPU Registers

OFFSET	NAME	DESCRIPTION
0h	REVID	Revision ID
4h	CONFIG	Configuration
10h	IRAWSTAT	Interrupt raw status/set
14h	IENSTAT	Interrupt enable status/clear
18h	IENSET	Interrupt enable
1Ch	IENCLR	Interrupt enable clear
20h	EOI	End of interrupt
200h	PROG0_MPSAR	Programmable range 0, start address
204h	PROG0_MPEAR	Programmable range 0, end address
208h	PROG0_MPPAR	Programmable range 0, memory page protection attributes
210h	PROG1_MPSAR	Programmable range 1, start address
214h	PROG1_MPEAR	Programmable range 1, end address
218h	PROG1_MPPAR	Programmable range 1, memory page protection attributes
220h	PROG2_MPSAR	Programmable range 2, start address
224h	PROG2_MPEAR	Programmable range 2, end address
228h	PROG2_MPPAR	Programmable range 2, memory page protection attributes

Table 6-8. MPU Registers (continued)

OFFSET	NAME	DESCRIPTION
230h	PROG3_MPSAR	Programmable range 3, start address
234h	PROG3_MPEAR	Programmable range 3, end address
238h	PROG3_MPPAR	Programmable range 3, memory page protection attributes
240h	PROG4_MPSAR	Programmable range 4, start address
244h	PROG4_MPEAR	Programmable range 4, end address
248h	PROG4_MPPAR	Programmable range 4, memory page protection attributes
250h	PROG5_MPSAR	Programmable range 5, start address
254h	PROG5_MPEAR	Programmable range 5, end address
258h	PROG5_MPPAR	Programmable range 5, memory page protection attributes
260h	PROG6_MPSAR	Programmable range 6, start address
264h	PROG6_MPEAR	Programmable range 6, end address
268h	PROG6_MPPAR	Programmable range 6, memory page protection attributes
270h	PROG7_MPSAR	Programmable range 7, start address
274h	PROG7_MPEAR	Programmable range 7, end address
278h	PROG7_MPPAR	Programmable range 7, memory page protection attributes
280h	PROG8_MPSAR	Programmable range 8, start address
284h	PROG8_MPEAR	Programmable range 8, end address
288h	PROG8_MPPAR	Programmable range 8, memory page protection attributes
290h	PROG9_MPSAR	Programmable range 9, start address
294h	PROG9_MPEAR	Programmable range 9, end address
298h	PROG9_MPPAR	Programmable range 9, memory page protection attributes
2A0h	PROG10_MPSAR	Programmable range 10, start address
2A4h	PROG10_MPEAR	Programmable range 10, end address
2A8h	PROG10_MPPAR	Programmable range 10, memory page protection attributes
2B0h	PROG11_MPSAR	Programmable range 11, start address
2B4h	PROG11_MPEAR	Programmable range 11, end address
2B8h	PROG11_MPPAR	Programmable range 11, memory page protection attributes
2C0h	PROG12_MPSAR	Programmable range 12, start address
2C4h	PROG12_MPEAR	Programmable range 12, end address
2C8h	PROG12_MPPAR	Programmable range 12, memory page protection attributes
2D0h	PROG13_MPSAR	Programmable range 13, start address
2D4h	PROG13_MPEAR	Programmable range 13, end address
2Dh	PROG13_MPPAR	Programmable range 13, memory page protection attributes
2E0h	PROG14_MPSAR	Programmable range 14, start address
2E4h	PROG14_MPEAR	Programmable range 14, end address
2E8h	PROG14_MPPAR	Programmable range 14, memory page protection attributes
2F0h	PROG15_MPSAR	Programmable range 15, start address
2F4h	PROG15_MPEAR	Programmable range 15, end address
2F8h	PROG15_MPPAR	Programmable range 15, memory page protection attributes
300h	FLTADDRR	Fault address
304h	FLTSTAT	Fault status
308h	FLTCLR	Fault clear

6.2.1.2 Device-Specific MPU Registers

6.2.1.2.1 Configuration Register (CONFIG)

The configuration register (CONFIG) contains the configuration value of the MPU.

Table 6-9. Configuration Register Field Descriptions

Bits	Field	Description
31-24	ADDR_WIDTH	Address alignment for range checking • 0 = 1KB alignment • 6 = 64KB alignment
23-20	NUM_FIXED	Number of fixed address ranges
19-16	NUM_PROG	Number of programmable address ranges
15-12	NUM_AIDS	Number of supported AIDs
11-1	Reserved	Reserved. Always read as 0.
0	ASSUME_ALLOWED	Assume allowed bit. When an address is not covered by any MPU protection range, this bit determines whether the transfer is assumed to be allowed or not. • 0 = Assume disallowed • 1 = Assume allowed

6.2.2 MPU Programmable Range Registers

6.2.2.1 Programmable Range *n* Start Address Register (PROG*n*_MPSAR)

The Programmable Address Start Register holds the start address for the range. This register is writeable by a supervisor entity only. If NS = 0 (non-secure mode) in the associated MPPAR register, then the register is also writeable only by a secure entity.

The start address must be aligned on a page boundary. The size of the page is 1K byte. The size of the page determines the width of the address field in MPSAR and MPEAR.

Figure 6-1. Programmable Range *n* Start Address Register (PROG*n*_MPSAR)

31	10	9	0
START_ADDR		Reserved	
R/W		R	

Legend: R = Read only; R/W = Read/Write

Table 6-10. Programmable Range *n* Start Address Register Field Descriptions

Bit	Field	Description
31-10	START_ADDR	Start address for range n
9-0	Reserved	Reserved. Always read as 0.

Table 6-11. MPU0-MPU5 Programmable Range *n* Start Address Register (PROG*n*_MPSAR) Reset Values

REGISTER	MPU0	MPU1	MPU2	MPU3	MPU4	MPU5
PROG0_MPSAR	0x01D0_0000	0x23A0_0000	0x02A0_0000	Reserved	Reserved	0x02A0_4000
PROG1_MPSAR	0x01F0_0000	0x23A0_2000	0x02A0_2000	Reserved	Reserved	0x02A0_5000
PROG2_MPSAR	0x02F0_0000	0x023A_6000	0x02A0_6000	Reserved	Reserved	0x02A0_6400
PROG3_MPSAR	0x0200_0000	0x23A0_6800	0x02A0_6800	Reserved	Reserved	0x02A0_7400
PROG4_MPSAR	0x020C_0000	0x23A0_7000	0x02A0_7000	Reserved	Reserved	0x02A0_A000
PROG5_MPSAR	0x021C_0000	0x23A0_8000	0x02A0_8000	Reserved	Reserved	0x02A0_D000
PROG6_MPSAR	0x021D_0000	0x23A0_C000	0x02A0_C000	Reserved	Reserved	0x02A0_E000
PROG7_MPSAR	0x021F_0000	0x23A0_E000	0x02A0_E000	Reserved	Reserved	0x02A0_F000
PROG8_MPSAR	0x0234_0000	0x23A0_F000	0x02A0_F000	Reserved	Reserved	0x02A0_F800
PROG9_MPSAR	0x0254_0000	0x23A0_F800	0x02A0_F800	Reserved	Reserved	0x02A1_2000
PROG10_MPSAR	0x0258_0000	0x23A1_0000	0x02A1_0000	Reserved	Reserved	0x02A1_C000
PROG11_MPSAR	0x0000_0000	0x23A1_C000	0x02A2_0000	Reserved	Reserved	0x02A2_8000
PROG12_MPSAR	0x0290_0000	0x23A4_0000	0x02A4_0000	Reserved	Reserved	0x02A6_0000
PROG13_MPSAR	0x01E8_0000	0x23A8_0000	0x02A8_0000	Reserved	Reserved	0x02AA_0000

Table 6-11. MPU0-MPU5 Programmable Range *n* Start Address Register (PROG*n*_MPSAR) Reset Values (continued)

REGISTER	MPU0	MPU1	MPU2	MPU3	MPU4	MPU5
PROG14_MPSAR	0x01E8_0800	0x23B0_0000	0x02AC_0000	Reserved	Reserved	0x02B0_0000
PROG15_MPSAR	0x01E0_0000	0x23B8_0000	0x02AE_0000	Reserved	Reserved	0x02B8_0000

Table 6-12. MPU6-MPU11 Programmable Range *n* Start Address Register (PROG*n*_MPSAR) Reset Values

REGISTER	MPU6	MPU7	MPU8	MPU9	MPU10	MPU11
PROG0_MPSAR	Reserved	0x2101_0000	0x3000_0000	0x0260_0000	0x0264_0000	0x0220_0000
PROG1_MPSAR	Reserved	0x0000_0000	0x3200_0000	0x0260_4000	0x0000_0000	0x0231_0000
PROG2_MPSAR	Reserved	0x0800_0000	0x3400_0000	0x0260_8000	N/A	0x0231_A000
PROG3_MPSAR	Reserved	0x1000_0000	0x3600_0000	0x0256_0000	N/A	0x0233_0000
PROG4_MPSAR	Reserved	0x1800_0000	0x3800_0000	0x0000_0000	N/A	0x0235_0000
PROG5_MPSAR	Reserved	0x2000_0000	0x3A00_0000	0x0000_0000	N/A	0x0263_0000
PROG6_MPSAR	Reserved	0x2800_0000	0x3C00_0000	0x0000_0000	N/A	0x0244_0000
PROG7_MPSAR	Reserved	0x3000_0000	0x2100_0800	0x0000_0000	N/A	0x024C_0000
PROG8_MPSAR	Reserved	0x3800_0000	N/A	0x0000_0000	N/A	0x0250_0000
PROG9_MPSAR	Reserved	0x4000_0000	N/A	0x0000_0000	N/A	0x0253_0000
PROG10_MPSAR	Reserved	0x4800_0000	N/A	0x0000_0000	N/A	0x0253_0C00
PROG11_MPSAR	Reserved	0x5000_0000	N/A	0x0000_0000	N/A	0x0260_B000
PROG12_MPSAR	Reserved	0x5800_0000	N/A	0x0000_0000	N/A	0x0262_0000
PROG13_MPSAR	Reserved	0x6000_0000	N/A	0x0000_0000	N/A	0x0300_0000
PROG14_MPSAR	Reserved	0x6800_0000	N/A	0x0000_0000	N/A	0x021E_0000
PROG15_MPSAR	Reserved	0x7000_0000	N/A	0x0000_0000	N/A	0x0268_0000

Table 6-13. MPU12-MPU15 Programmable Range *n* Start Address Register (PROG*n*_MPSAR) Reset Values

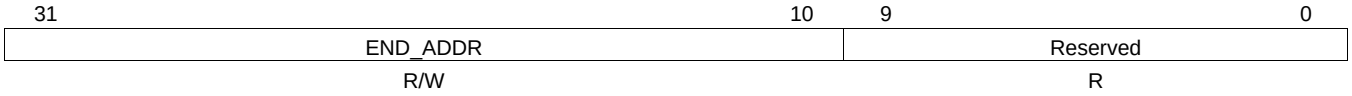
REGISTER	MPU12	MPU13	MPU14	MPU15
PROG0_MPSAR	0x2100_0400	0x2100_0400	0x2100_0800	0x2400_0000
PROG1_MPSAR	0x0000_0000	0x0000_0000	0x0000_0000	0x2408_0000
PROG2_MPSAR	N/A	N/A	N/A	0x2410_0000
PROG3_MPSAR	N/A	N/A	N/A	0x2500_0000
PROG4_MPSAR	N/A	N/A	N/A	0x0000_0000
PROG5_MPSAR	N/A	N/A	N/A	0x0000_0000
PROG6_MPSAR	N/A	N/A	N/A	0x0000_0000
PROG7_MPSAR	N/A	N/A	N/A	0x0000_0000
PROG8_MPSAR	N/A	N/A	N/A	N/A
PROG9_MPSAR	N/A	N/A	N/A	N/A
PROG10_MPSAR	N/A	N/A	N/A	N/A
PROG11_MPSAR	N/A	N/A	N/A	N/A
PROG12_MPSAR	N/A	N/A	N/A	N/A
PROG13_MPSAR	N/A	N/A	N/A	N/A
PROG14_MPSAR	N/A	N/A	N/A	N/A
PROG15_MPSAR	N/A	N/A	N/A	N/A

6.2.2.2 Programmable Range *n* - End Address Register (PROG*n*_MPEAR)

The programmable address end register holds the end address for the range. This register is writeable by a supervisor entity only. If NS = 0 (non-secure mode) in the associated MPPAR register then the register is also writeable only by a secure entity.

The end address must be aligned on a page boundary. The size of the page depends on the MPU number. The page size for MPU1 is 1K byte and for MPU2 it is 64K bytes. The size of the page determines the width of the address field in MPSAR and MPEAR.

Figure 6-2. Programmable Range *n* End Address Register (PROG*n*_MPEAR)



Legend: R = Read only; R/W = Read/Write

Table 6-14. Programmable Range *n* End Address Register Field Descriptions

Bit	Field	Description
31-10	END_ADDR	End address for range n
9-0	Reserved	Reserved. Always read as 3FFh.

Table 6-15. MPU0-MPU5 Programmable Range *n* End Address Register (PROG*n*_MPEAR) Reset Values

REGISTER	MPU0	MPU1	MPU2	MPU3	MPU4	MPU5
PROG0_MPEAR	0x01DF_FFFF	0x23A0_1FFF	0x02A0_00FF	Reserved	Reserved	0x02A0_4FFF
PROG1_MPEAR	0x01F7_FFFF	0x23A0_5FFF	0x02A0_3FFF	Reserved	Reserved	0x02A0_5FFF
PROG2_MPEAR	0x02FF_FFFF	0x23A0_67FF	0x02A0_63FF	Reserved	Reserved	0x02A0_67FF
PROG3_MPEAR	0x020B_FFFF	0x23A0_6FFF	0x02A0_6FFF	Reserved	Reserved	0x02A0_7FFF
PROG4_MPEAR	0x020F_FFFF	0x23A0_7FFF	0x02A0_73FF	Reserved	Reserved	0x02A0_BFFF
PROG5_MPEAR	0x021C_83FF	0x23A0_BFFF	0x02A0_9FFF	Reserved	Reserved	0x02A0_DFFF
PROG6_MPEAR	0x021D_C0FF	0x23A0_DFFF	0x02A0_CFFF	Reserved	Reserved	0x02A0_E7FF
PROG7_MPEAR	0x021F_C7FF	0x23A0_EFFF	0x02A0_E7FF	Reserved	Reserved	0x02A0_F7FF
PROG8_MPEAR	0x0234_C0FF	0x23A0_F7FF	0x02A0_F7FF	Reserved	Reserved	0x02A0_FFFF
PROG9_MPEAR	0x0255_FFFF	0x23A0_FFFF	0x02A0_FFFF	Reserved	Reserved	0x02A1_7FFF
PROG10_MPEAR	0x025F_FFFF	0x23A1_BFFF	0x02A1_1FFF	Reserved	Reserved	0x02A1_FFFF
PROG11_MPEAR	0x0000_0000	0x23A3_FFFF	0x02A2_5FFF	Reserved	Reserved	0x02A3_FFFF
PROG12_MPEAR	0x029F_FFFF	0x23A7_FFFF	0x02A5_FFFF	Reserved	Reserved	0x02A7_FFFF
PROG13_MPEAR	0x01E8_07FF	0x23AF_FFFF	0x02A9_FFFF	Reserved	Reserved	0x02AB_FFFF
PROG14_MPEAR	0x01E8_43FF	0x23B7_FFFF	0x02AD_FFFF	Reserved	Reserved	0x02B7_FFFF
PROG15_MPEAR	0x01E7_FFFF	0x23BF_FFFF	0x02AF_FFFF	Reserved	Reserved	0x02BF_FFFF

Table 6-16. MPU6-MPU11 Programmable Range *n* End Address Register (PROG*n*_MPEAR) Reset Values

REGISTER	MPU6	MPU7	MPU8	MPU9	MPU10	MPU11
PROG0_MPEAR	Reserved	0x2103_FFFF	0x31FF_FFFF	0x0260_1FFF	0x0264_07FF	0x022F_027F
PROG1_MPEAR	Reserved	0x07FF_FFFF	0x33FF_FFFF	0x0260_5FFF	0x0000_0000	0x0231_01FF
PROG2_MPEAR	Reserved	0x0FFF_FFFF	0x35FF_FFFF	0x0260_9FFF	N/A	0x0232_FFFF
PROG3_MPEAR	Reserved	0x17FF_FFFF	0x37FF_FFFF	0x0257_FFFF	N/A	0x0233_07FF
PROG4_MPEAR	Reserved	0x1FFF_FFFF	0x39FF_FFFF	0x0000_0000	N/A	0x0235_0FFF
PROG5_MPEAR	Reserved	0x27FF_FFFF	0x3BFF_FFFF	0x0000_0000	N/A	0x0263_FFFF
PROG6_MPEAR	Reserved	0x2FFF_FFFF	0x3FFF_FFFF	0x0000_0000	N/A	0x024B_3FFF
PROG7_MPEAR	Reserved	0x37FF_FFFF	0x2100_0AFF	0x0000_0000	N/A	0x024C_0BFF
PROG8_MPEAR	Reserved	0x3FFF_FFFF	N/A	0x0000_0000	N/A	0x0250_7FFF
PROG9_MPEAR	Reserved	0x47FF_FFFF	N/A	0x0000_0000	N/A	0x0253_0BFF
PROG10_MPEAR	Reserved	0x4FFF_FFFF	N/A	0x0000_0000	N/A	0x0253_FFFF
PROG11_MPEAR	Reserved	0x57FF_FFFF	N/A	0x0000_0000	N/A	0x0260_BFFF
PROG12_MPEAR	Reserved	0x5FFF_FFFF	N/A	0x0000_0000	N/A	0x0262_0FFF

Table 6-16. MPU6-MPU11 Programmable Range *n* End Address Register (PROG*n*_MPEAR) Reset Values (continued)

REGISTER	MPU6	MPU7	MPU8	MPU9	MPU10	MPU11
PROG13_MPEAR	Reserved	0x67FF_FFFF	N/A	0x0000_0000	N/A	0x03FF_FFFF
PROG14_MPEAR	Reserved	0x6FFF_FFFF	N/A	0x0000_0000	N/A	0x021E_1FFF
PROG15_MPEAR	Reserved	0x7FFF_FFFF	N/A	0x0000_0000	N/A	0x026F_FFFF

Table 6-17. MPU12-MPU15 Programmable Range *n* End Address Register (PROG*n*_MPEAR) Reset Values

REGISTER	MPU12	MPU13	MPU14	MPU15
PROG0_MPEAR	0x2100_07FF	0x2100_07FF	0x2100_0AFF	0x2407_FFFF
PROG1_MPEAR	0x0000_0000	0x0000_0000	0x0000_0000	0x240F_FFFF
PROG2_MPEAR	N/A	N/A	N/A	0x24FF_FFFF
PROG3_MPEAR	N/A	N/A	N/A	0x2507_FFFF
PROG4_MPEAR	N/A	N/A	N/A	0x2508FFFF
PROG5_MPEAR	N/A	N/A	N/A	0x0000_0000
PROG6_MPEAR	N/A	N/A	N/A	0x0000_0000
PROG7_MPEAR	N/A	N/A	N/A	0x0000_0000
PROG8_MPEAR	N/A	N/A	N/A	N/A
PROG9_MPEAR	N/A	N/A	N/A	N/A
PROG10_MPEAR	N/A	N/A	N/A	N/A
PROG11_MPEAR	N/A	N/A	N/A	N/A
PROG12_MPEAR	N/A	N/A	N/A	N/A
PROG13_MPEAR	N/A	N/A	N/A	N/A
PROG14_MPEAR	N/A	N/A	N/A	N/A
PROG15_MPEAR	N/A	N/A	N/A	N/A

6.2.2.3 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPAR)

The programmable address memory protection page attribute register holds the permissions for the region. This register is writeable only by a non-debug supervisor entity. If NS = 0 (secure mode) then the register is also writeable only by a non-debug secure entity. The NS bit is writeable only by a non-debug secure entity. For debug accesses, the register is writeable only when NS = 1 or EMU = 1.

Figure 6-3. Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPAR)

31	26	25	24	23	22	21	20	19	18	17	16	15
Reserved		AID15	AID14	AID13	AID12	AID11	AID10	AID9	AID8	AID7	AID6	AID5
R		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AID4	AID3	AID2	AID1	AID0	AIDX	Reserved	NS	EMU	SR	SW	SX	UR	UW	UX
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Legend: R = Read only; R/W = Read/Write

Table 6-18. Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions

Bits	Name	Description
31-26	Reserved	Reserved. Always read as 0.
25	AID15	Controls access from ID = 15 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
24	AID14	Controls access from ID = 14 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions

Table 6-18. Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions (continued)

Bits	Name	Description
23	AID13	Controls access from ID = 13 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
22	AID12	Controls access from ID = 12 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
21	AID11	Controls access from ID = 11 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
20	AID10	Controls access from ID = 10 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
19	AID9	Controls access from ID = 9 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
18	AID8	Controls access from ID = 8 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
17	AID7	Controls access from ID = 7 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
16	AID6	Controls access from ID = 6 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
15	AID5	Controls access from ID = 5 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
14	AID4	Controls access from ID = 4 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
13	AID3	Controls access from ID = 3 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
12	AID2	Controls access from ID = 2 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
11	AID1	Controls access from ID = 1 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
10	AID0	Controls access from ID = 0 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
9	AIDX	Controls access from ID > 15 • 0 = Access is not checked for permissions • 1 = Access is checked for permissions
8	Reserved	Reserved. Always reads as 0.
7	NS	Non-secure access permission • 0 = Only secure access allowed • 1 = Non-secure access allowed

Table 6-18. Programmable Range *n* Memory Protection Page Attribute Register Field Descriptions (continued)

Bits	Name	Description
6	EMU	Emulation (debug) access permission. This bit is ignored if NS = 1 • 0 = Debug access not allowed • 1 = Debug access allowed
5	SR	Supervisor Read permission • 0 = Access not allowed • 1 = Access allowed
4	SW	Supervisor Write permission • 0 = Access not allowed • 1 = Access allowed
3	SX	Supervisor Execute permission • 0 = Access not allowed • 1 = Access allowed
2	UR	User Read permission • 0 = Access not allowed • 1 = Access allowed
1	UW	User Write permission • 0 = Access not allowed • 1 = Access allowed
0	UX	User Execute permission • 0 = Access not allowed • 1 = Access allowed

Table 6-19. MPU0-MPU5 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPAR) Reset Values

REGISTER	MPU0	MPU1	MPU2	MPU3	MPU4	MPU5
PROG0_MPPAR	0x03FF_FCB6	0x03FF_FCB6	0x03FF_FCB6	Reserved	Reserved	0x03FF_FCB4
PROG1_MPPAR	0x03FF_FCB6	0x03FF_FCB4	0x03FF_FCB4	Reserved	Reserved	0x03FF_FCB4
PROG2_MPPAR	0x03FF_FCB6	0x03FF_FCA4	0x03FF_FCA4	Reserved	Reserved	0x03FF_FCA4
PROG3_MPPAR	0x03FF_FCB6	0x03FF_FCB4	0x03FF_FCB4	Reserved	Reserved	0x03FF_FCF4
PROG4_MPPAR	0x03FF_FCB6	0x03FF_FCF4	0x03FF_FCF4	Reserved	Reserved	0x03FF_FCB4
PROG5_MPPAR	0x03FF_FCB6	0x03FF_FCB4	0x03FF_FCB4	Reserved	Reserved	0x03FF_FCB4
PROG6_MPPAR	0x03FF_FCB6	0x03FF_FCB4	0x03FF_FCB4	Reserved	Reserved	0x03FF_FCB4
PROG7_MPPAR	0x03FF_FCB6	0x03FF_FCB4	0x03FF_FCB4	Reserved	Reserved	0x03FF_FCB4
PROG8_MPPAR	0x03FF_FCB6	0x03FF_FCB4	0x03FF_FCB4	Reserved	Reserved	0x03FF_FCF4
PROG9_MPPAR	0x03FF_FCB6	0x03FF_FCF4	0x03FF_FCF4	Reserved	Reserved	0x03FF_FCB4
PROG10_MPPAR	0x03FF_FCB6	0x03FF_FCB4	0x03FF_FCB4	Reserved	Reserved	0x03FF_FCF4
PROG11_MPPAR	0x03FF_FCB6	0x03FF_FCF4	0x03FF_FCF4	Reserved	Reserved	0x03FF_FCF4
PROG12_MPPAR	0x03FF_FCB4	0x03FF_FCA4	0x03FF_FCA4	Reserved	Reserved	0x03FF_FCA4
PROG13_MPPAR	0x03FF_FCB6	0x03FF_FCB6	0x03FF_FCB6	Reserved	Reserved	0x03FF_FCB6
PROG14_MPPAR	0x03FF_FCB0	0x03FF_FCA4	0x03FF_FCB6	Reserved	Reserved	0x03FF_FCA4
PROG15_MPPAR	0x03FF_FCB6	0x03FF_FCA4	0x03FF_FCB6	Reserved	Reserved	0x03FF_FCA4

Table 6-20. MPU6-MPU11 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPAR) Reset Values

REGISTER	MPU6	MPU7	MPU8	MPU9	MPU10	MPU11
PROG0_MPPAR	Reserved	0x03FF_FCB6	0x03FF_FCBF	0x03FF_FCB6	0x03FF_FCB6	0x03FF_FCB6
PROG1_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6	0x03FF_FCB6	0x03FF_FCB0
PROG2_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6	N/A	0x03FF_FCB6

Table 6-20. MPU6-MPU11 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPAR) Reset Values (continued)

REGISTER	MPU6	MPU7	MPU8	MPU9	MPU10	MPU11
PROG3_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6	N/A	0x03FF_FCB0
PROG4_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6	N/A	0x03FF_FCB0
PROG5_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6	N/A	0x03FF_FCB6
PROG6_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6	N/A	0x03FF_FCB6
PROG7_MPPAR	Reserved	0x03FF_FCBF	0x03FF_FCB6	0x03FF_FCB6	N/A	0x03FF_FCB0
PROG8_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB0
PROG9_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB6
PROG10_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB6
PROG11_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB6
PROG12_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB0
PROG13_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB6
PROG14_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB0
PROG15_MPPAR	Reserved	0x03FF_FCBF	N/A	0x03FF_FCB6	N/A	0x03FF_FCB6

Table 6-21. MPU12-MPU15 Programmable Range *n* Memory Protection Page Attribute Register (PROG*n*_MPPAR) Reset Values

REGISTER	MPU12	MPU13	MPU14	MPU15
PROG0_MPPAR	0x03FF_FCB6	0x03FF_FCB6	0x03FF_FCB6	0x03FF_FCB6
PROG1_MPPAR	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCBF	0x03FF_FCB6
PROG2_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG3_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG4_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG5_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG6_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG7_MPPAR	N/A	N/A	N/A	0x03FF_FCB6
PROG8_MPPAR	N/A	N/A	N/A	N/A
PROG9_MPPAR	N/A	N/A	N/A	N/A
PROG10_MPPAR	N/A	N/A	N/A	N/A
PROG11_MPPAR	N/A	N/A	N/A	N/A
PROG12_MPPAR	N/A	N/A	N/A	N/A
PROG13_MPPAR	N/A	N/A	N/A	N/A
PROG14_MPPAR	N/A	N/A	N/A	N/A
PROG15_MPPAR	N/A	N/A	N/A	N/A

6.3 Interrupts for AM5K2E0x

This section discusses the interrupt sources, controller, and topology. Also provided are tables describing the interrupt events.

6.3.1 Interrupt Sources and Interrupt Controller

The ARM CorePac interrupts on the AM5K2E0x device are configured through the ARM CorePac Interrupt Controller. It allows for up to 480 system events to be programmed to any of the ARM core's IRQ/FIQ interrupts. In addition error-class events or infrequently used events are also routed through the system event router to offload the ARM CorePac interrupt controller. This is accomplished through the CorePac Interrupt Controller block CIC2. Further, CIC2 provides 8 events each to EDMA3CC0, EDMA3CC1, EDMA3C2, EDMA3CC3, EDMA3CC4, and Hyperlink.

Modules such as CP_MPU, BOOT_CFG, and CP_Tracer have level interrupts and EOI handshaking interface. The EOI value is 0 for CP_MPU, BOOT_CFG, and CP_Tracer.

Figure 6-4 shows the AM5K2E0x interrupt topology.

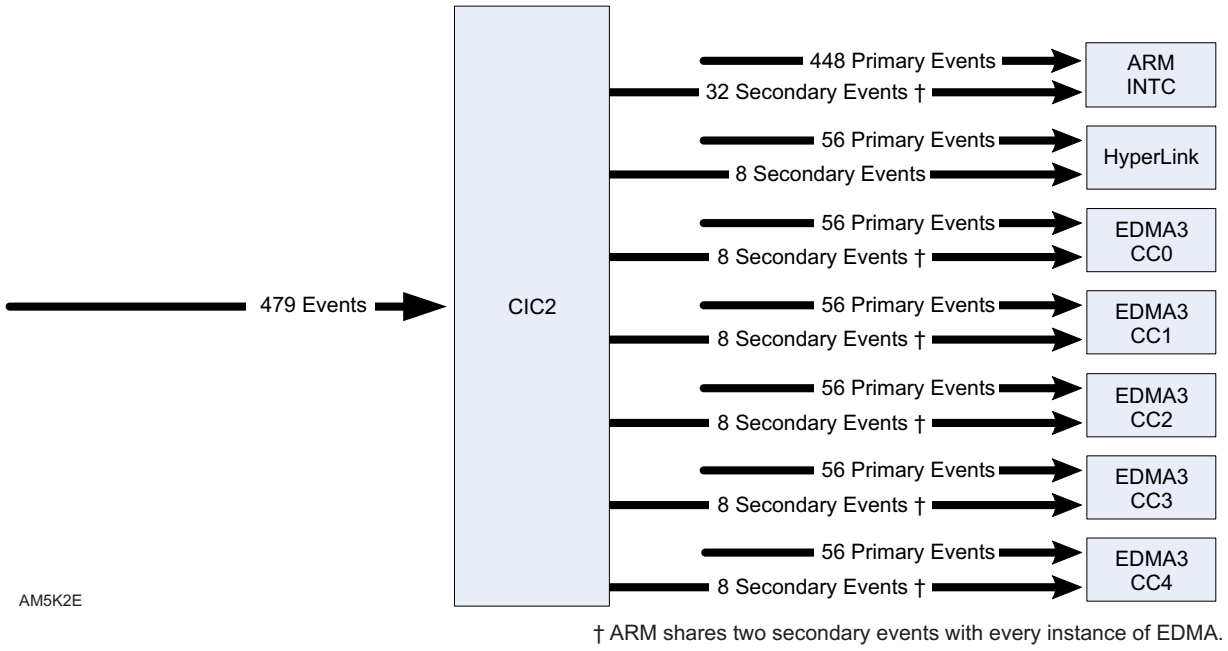


Figure 6-4. Interrupt Topology

Table 6-22 lists the ARM CorePac event inputs

Table 6-22. System Event Mapping — ARM CorePac Interrupts

EVENT NO.	EVENT NAME	DESCRIPTION
0	RSTMUX_INT8	Boot config watchdog timer expiration (timer 16) event for ARM core 0
1	RSTMUX_INT9	Boot config watchdog timer expiration (timer 17) event for ARM core 1
2	RSTMUX_INT10	Boot config watchdog timer expiration (timer 18) event for ARM core 2
3	RSTMUX_INT11	Boot config watchdog timer expiration (timer 19) event for ARM core 3
4	IPC_GR8	Boot config IPCG
5	IPC_GR9	Boot config IPCG
6	IPC_GR10	Boot config IPCG
7	IPC_GR11	Boot config IPCG
8	SEM_INT8	Semaphore interrupt
9	SEM_INT9	Semaphore interrupt
10	SEM_INT10	Semaphore interrupt
11	SEM_INT11	Semaphore interrupt
12	SEM_ERR8	Semaphore error interrupt
13	SEM_ERR9	Semaphore error interrupt
14	SEM_ERR10	Semaphore error interrupt
15	SEM_ERR11	Semaphore error interrupt
16	MSMC_MPF_ERROR8	Memory protection fault indicators for system master PrivID = 8
17	MSMC_MPF_ERROR9	Memory protection fault indicators for system master PrivID = 9
18	MSMC_MPF_ERROR10	Memory protection fault indicators for system master PrivID = 10
19	MSMC_MPF_ERROR11	Memory protection fault indicators for system master PrivID = 11

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
20	ARM_NPMUIRQ0	ARM performance monitoring unit interrupt request
21	ARM_NPMUIRQ1	ARM performance monitoring unit interrupt request
22	ARM_NPMUIRQ2	ARM performance monitoring unit interrupt request
23	ARM_NPMUIRQ3	ARM performance monitoring unit interrupt request
24	ARM_NINTERRIRQ	ARM internal memory ECC error interrupt request
25	ARM_NAXIERRIRQ	ARM bus error interrupt request
26	PCIE_0_INT0	PCIE0 legacy INTA interrupt
27	PCIE_0_INT1	PCIE0 legacy INTB interrupt
28	PCIE_0_INT2	PCIE0 legacy INTC interrupt
29	PCIE_0_INT3	PCIE0 legacy INTD interrupt
30	PCIE_0_INT4	PCIE0 MSI interrupt
31	PCIE_0_INT5	PCIE0 MSI interrupt
32	PCIE_0_INT6	PCIE0 MSI interrupt
33	PCIE_0_INT7	PCIE0 MSI interrupt
34	PCIE_0_INT8	PCIE0 MSI interrupt
35	PCIE_0_INT9	PCIE0 MSI interrupt
36	PCIE_0_INT10	PCIE0 MSI interrupt
37	PCIE_0_INT11	PCIE0 MSI interrupt
38	PCIE_0_INT12	PCIE0 error interrupt
39	PCIE_0_INT13	PCIE0 power management interrupt
40	QMSS_QUE_PEND_658	Navigator transmit queue pending event for indicated queue
41	QMSS_QUE_PEND_659	Navigator transmit queue pending event for indicated queue
42	QMSS_QUE_PEND_660	Navigator transmit queue pending event for indicated queue
43	QMSS_QUE_PEND_661	Navigator transmit queue pending event for indicated queue
44	QMSS_QUE_PEND_662	Navigator transmit queue pending event for indicated queue
45	QMSS_QUE_PEND_663	Navigator transmit queue pending event for indicated queue
46	QMSS_QUE_PEND_664	Navigator transmit queue pending event for indicated queue
47	QMSS_QUE_PEND_665	Navigator transmit queue pending event for indicated queue
48	QMSS_QUE_PEND_528	Navigator transmit queue pending event for indicated queue
49	QMSS_QUE_PEND_529	Navigator transmit queue pending event for indicated queue
50	QMSS_QUE_PEND_530	Navigator transmit queue pending event for indicated queue
51	QMSS_QUE_PEND_531	Navigator transmit queue pending event for indicated queue
52	QMSS_QUE_PEND_532	Navigator transmit queue pending event for indicated queue
53	QMSS_QUE_PEND_533	Navigator transmit queue pending event for indicated queue
54	QMSS_QUE_PEND_534	Navigator transmit queue pending event for indicated queue
55	QMSS_QUE_PEND_535	Navigator transmit queue pending event for indicated queue
56	QMSS_QUE_PEND_536	Navigator transmit queue pending event for indicated queue
57	QMSS_QUE_PEND_537	Navigator transmit queue pending event for indicated queue
58	QMSS_QUE_PEND_538	Navigator transmit queue pending event for indicated queue
59	QMSS_QUE_PEND_539	Navigator transmit queue pending event for indicated queue
60	QMSS_QUE_PEND_540	Navigator transmit queue pending event for indicated queue
61	QMSS_QUE_PEND_541	Navigator transmit queue pending event for indicated queue
62	QMSS_QUE_PEND_542	Navigator transmit queue pending event for indicated queue
63	QMSS_QUE_PEND_543	Navigator transmit queue pending event for indicated queue
64	QMSS_QUE_PEND_544	Navigator transmit queue pending event for indicated queue
65	QMSS_QUE_PEND_545	Navigator transmit queue pending event for indicated queue
66	QMSS_QUE_PEND_546	Navigator transmit queue pending event for indicated queue

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
67	QMSS_QUE_PEND_547	Navigator transmit queue pending event for indicated queue
68	QMSS_QUE_PEND_548	Navigator transmit queue pending event for indicated queue
69	QMSS_QUE_PEND_549	Navigator transmit queue pending event for indicated queue
70	QMSS_QUE_PEND_550	Navigator transmit queue pending event for indicated queue
71	QMSS_QUE_PEND_551	Navigator transmit queue pending event for indicated queue
72	QMSS_QUE_PEND_552	Navigator transmit queue pending event for indicated queue
73	QMSS_QUE_PEND_553	Navigator transmit queue pending event for indicated queue
74	QMSS_QUE_PEND_554	Navigator transmit queue pending event for indicated queue
75	QMSS_QUE_PEND_555	Navigator transmit queue pending event for indicated queue
76	QMSS_QUE_PEND_556	Navigator transmit queue pending event for indicated queue
77	QMSS_QUE_PEND_557	Navigator transmit queue pending event for indicated queue
78	QMSS_QUE_PEND_558	Navigator transmit queue pending event for indicated queue
79	QMSS_QUE_PEND_559	Navigator transmit queue pending event for indicated queue
80	Reserved	Reserved
81	Reserved	Reserved
82	USIM_PONIRQ	USIM interrupt
83	USIM_RREQ	USIM read DMA event
84	USIM_WREQ	USIM write DMA event
85	TSIP_RCV_FINT0	TSIP receive frame interrupt for channel 0
86	TSIP_XMT_FINT0	TSIP transmit frame interrupt for channel 0
87	TSIP_RCV_SFINT0	TSIP receive super frame interrupt for channel 0
88	TSIP_XMT_SFINT0	TSIP transmit super frame interrupt for channel 0
89	TSIP_EINT0	TSIP error interrupt for channel 0
90	TSIP_RCV_FINT1	TSIP receive frame interrupt for channel 1
91	TSIP_XMT_FINT1	TSIP transmit frame interrupt for channel 1
92	TSIP_RCV_SFINT1	TSIP receive super frame interrupt for channel 1
93	TSIP_XMT_SFINT1	TSIP transmit super frame interrupt for channel 1
94	TSIP_EINT1	TSIP error interrupt for channel 1
95	Reserved	Reserved
96	TIMER_8_INTL	Timer interrupt low
97	TIMER_8_INTH	Timer interrupt high
98	TIMER_9_INTL	Timer interrupt low
99	TIMER_9_INTH	Timer interrupt high
100	TIMER_10_INTL	Timer interrupt low
101	TIMER_10_INTH	Timer interrupt high
102	TIMER_11_INTL	Timer interrupt low
103	TIMER_11_INTH	Timer interrupt high
104	TIMER_12_INTL	Timer interrupt low
105	TIMER_12_INTH	Timer interrupt high
106	TIMER_13_INTL	Timer interrupt low
107	TIMER_13_INTH	Timer interrupt high
108	TIMER_14_INTL	Timer interrupt low
109	TIMER_14_INTH	Timer interrupt high
110	TIMER_15_INTL	Timer interrupt low
111	TIMER_15_INTH	Timer interrupt high
112	TIMER_16_INTL	Timer interrupt low
113	TIMER_16_INTH	Timer interrupt high

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
114	TIMER_17_INTL	Timer interrupt low
115	TIMER_17_INTH	Timer interrupt high
116	TIMER_18_INTL	Timer interrupt low
117	TIMER_18_INTH	Timer interrupt high
118	TIMER_19_INTL	Timer interrupt low
119	TIMER_19_INTH	Timer interrupt high
120	GPIO_INT0	GPIO interrupt
121	GPIO_INT1	GPIO interrupt
122	GPIO_INT2	GPIO interrupt
123	GPIO_INT3	GPIO interrupt
124	GPIO_INT4	GPIO interrupt
125	GPIO_INT5	GPIO interrupt
126	GPIO_INT6	GPIO interrupt
127	GPIO_INT7	GPIO interrupt
128	GPIO_INT8	GPIO interrupt
129	GPIO_INT9	GPIO interrupt
130	GPIO_INT10	GPIO interrupt
131	GPIO_INT11	GPIO interrupt
132	GPIO_INT12	GPIO interrupt
133	GPIO_INT13	GPIO interrupt
134	GPIO_INT14	GPIO interrupt
135	GPIO_INT15	GPIO interrupt
136	GPIO_INT16	GPIO interrupt
137	GPIO_INT17	GPIO interrupt
138	GPIO_INT18	GPIO interrupt
139	GPIO_INT19	GPIO interrupt
140	GPIO_INT20	GPIO interrupt
141	GPIO_INT21	GPIO interrupt
142	GPIO_INT22	GPIO interrupt
143	GPIO_INT23	GPIO interrupt
144	GPIO_INT24	GPIO interrupt
145	GPIO_INT25	GPIO interrupt
146	GPIO_INT26	GPIO interrupt
147	GPIO_INT27	GPIO interrupt
148	GPIO_INT28	GPIO interrupt
149	GPIO_INT29	GPIO interrupt
150	GPIO_INT30	GPIO interrupt
151	GPIO_INT31	GPIO interrupt
152	USB_0_INT00	USB 0 event ring 0 interrupt
153	USB_0_INT01	USB 0 event ring 1 interrupt
154	USB_0_INT02	USB 0 event ring 2 interrupt
155	USB_0_INT03	USB 0 event ring 3 interrupt
156	USB_0_INT04	USB 0 event ring 4 interrupt
157	USB_0_INT05	USB 0 event ring 5 interrupt
158	USB_0_INT06	USB 0 event ring 6 interrupt
159	USB_0_INT07	USB 0 event ring 7 interrupt
160	USB_0_INT08	USB 0 event ring 8 interrupt

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
161	USB_0_INT09	USB 0 event ring 9 interrupt
162	USB_0_INT10	USB 0 event ring 10 interrupt
163	USB_0_INT11	USB 0 event ring 11 interrupt
164	USB_0_INT12	USB 0 event ring 12 interrupt
165	USB_0_INT13	USB 0 event ring 13 interrupt
166	USB_0_INT14	USB 0 event ring 14 interrupt
167	USB_0_INT15	USB 0 event ring 15 interrupt
168	USB_0_OABSINT	USB 0 OABS interrupt
169	USB_0_MISCINT	USB0_misc_int
170	MSMC_DEDC_CERROR	MSMC interrupt
171	MSMC_DEDC_NC_ERROR	MSMC interrupt
172	MSMC_DEDC_SCRUB_CERROR	MSMC interrupt
173	MSMC_DEDC_SCRUB_NC_ERROR	MSMC interrupt
174	Reserved	Reserved
175	Reserved	Reserved
176	QMSS1_ECC_INTR	Navigator ECC error interrupt
177	QMSS_INTD_1_PKTDMA_0	Navigator interrupt for Packet DMA starvation
178	QMSS_INTD_1_PKTDMA_1	Navigator interrupt for Packet DMA starvation
179	QMSS_INTD_1_HIGH_0	Navigator hi interrupt
180	QMSS_INTD_1_HIGH_1	Navigator hi interrupt
181	QMSS_INTD_1_HIGH_2	Navigator hi interrupt
182	QMSS_INTD_1_HIGH_3	Navigator hi interrupt
183	QMSS_INTD_1_HIGH_4	Navigator hi interrupt
184	QMSS_INTD_1_HIGH_5	Navigator hi interrupt
185	QMSS_INTD_1_HIGH_6	Navigator hi interrupt
186	QMSS_INTD_1_HIGH_7	Navigator hi interrupt
187	QMSS_INTD_1_HIGH_8	Navigator hi interrupt
188	QMSS_INTD_1_HIGH_9	Navigator hi interrupt
189	QMSS_INTD_1_HIGH_10	Navigator hi interrupt
190	QMSS_INTD_1_HIGH_11	Navigator hi interrupt
191	QMSS_INTD_1_HIGH_12	Navigator hi interrupt
192	QMSS_INTD_1_HIGH_13	Navigator hi interrupt
193	QMSS_INTD_1_HIGH_14	Navigator hi interrupt
194	QMSS_INTD_1_HIGH_15	Navigator hi interrupt
195	QMSS_INTD_1_HIGH_16	Navigator hi interrupt
196	QMSS_INTD_1_HIGH_17	Navigator hi interrupt
197	QMSS_INTD_1_HIGH_18	Navigator hi interrupt
198	QMSS_INTD_1_HIGH_19	Navigator hi interrupt
199	QMSS_INTD_1_HIGH_20	Navigator hi interrupt
200	QMSS_INTD_1_HIGH_21	Navigator hi interrupt
201	QMSS_INTD_1_HIGH_22	Navigator hi interrupt
202	QMSS_INTD_1_HIGH_23	Navigator hi interrupt
203	QMSS_INTD_1_HIGH_24	Navigator hi interrupt
204	QMSS_INTD_1_HIGH_25	Navigator hi interrupt
205	QMSS_INTD_1_HIGH_26	Navigator hi interrupt
206	QMSS_INTD_1_HIGH_27	Navigator hi interrupt
207	QMSS_INTD_1_HIGH_28	Navigator hi interrupt

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
208	QMSS_INTD_1_HIGH_29	Navigator hi interrupt
209	QMSS_INTD_1_HIGH_30	Navigator hi interrupt
210	QMSS_INTD_1_HIGH_31	Navigator hi interrupt
211	QMSS_INTD_1_LOW_0	Navigator interrupt
212	QMSS_INTD_1_LOW_1	Navigator interrupt
213	QMSS_INTD_1_LOW_2	Navigator interrupt
214	QMSS_INTD_1_LOW_3	Navigator interrupt
215	QMSS_INTD_1_LOW_4	Navigator interrupt
216	QMSS_INTD_1_LOW_5	Navigator interrupt
217	QMSS_INTD_1_LOW_6	Navigator interrupt
218	QMSS_INTD_1_LOW_7	Navigator interrupt
219	QMSS_INTD_1_LOW_8	Navigator interrupt
220	QMSS_INTD_1_LOW_9	Navigator interrupt
221	QMSS_INTD_1_LOW_10	Navigator interrupt
222	QMSS_INTD_1_LOW_11	Navigator interrupt
223	QMSS_INTD_1_LOW_12	Navigator interrupt
224	QMSS_INTD_1_LOW_13	Navigator interrupt
225	QMSS_INTD_1_LOW_14	Navigator interrupt
226	QMSS_INTD_1_LOW_15	Navigator interrupt
227	Reserved	Reserved
228	Reserved	Reserved
229	QMSS_INTD_2_HIGH_0	Navigator second hi interrupt
230	QMSS_INTD_2_HIGH_1	Navigator second hi interrupt
231	QMSS_INTD_2_HIGH_2	Navigator second hi interrupt
232	QMSS_INTD_2_HIGH_3	Navigator second hi interrupt
233	QMSS_INTD_2_HIGH_4	Navigator second hi interrupt
234	QMSS_INTD_2_HIGH_5	Navigator second hi interrupt
235	QMSS_INTD_2_HIGH_6	Navigator second hi interrupt
236	QMSS_INTD_2_HIGH_7	Navigator second hi interrupt
237	QMSS_INTD_2_HIGH_8	Navigator second hi interrupt
238	QMSS_INTD_2_HIGH_9	Navigator second hi interrupt
239	QMSS_INTD_2_HIGH_10	Navigator second hi interrupt
240	QMSS_INTD_2_HIGH_11	Navigator second hi interrupt
241	QMSS_INTD_2_HIGH_12	Navigator second hi interrupt
242	QMSS_INTD_2_HIGH_13	Navigator second hi interrupt
243	QMSS_INTD_2_HIGH_14	Navigator second hi interrupt
244	QMSS_INTD_2_HIGH_15	Navigator second hi interrupt
245	QMSS_INTD_2_HIGH_16	Navigator second hi interrupt
246	QMSS_INTD_2_HIGH_17	Navigator second hi interrupt
247	QMSS_INTD_2_HIGH_18	Navigator second hi interrupt
248	QMSS_INTD_2_HIGH_19	Navigator second hi interrupt
249	QMSS_INTD_2_HIGH_20	Navigator second hi interrupt
250	QMSS_INTD_2_HIGH_21	Navigator second hi interrupt
251	QMSS_INTD_2_HIGH_22	Navigator second hi interrupt
252	QMSS_INTD_2_HIGH_23	Navigator second hi interrupt
253	QMSS_INTD_2_HIGH_24	Navigator second hi interrupt
254	QMSS_INTD_2_HIGH_25	Navigator second hi interrupt

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
255	QMSS_INTD_2_HIGH_26	Navigator second hi interrupt
256	QMSS_INTD_2_HIGH_27	Navigator second hi interrupt
257	QMSS_INTD_2_HIGH_28	Navigator second hi interrupt
258	QMSS_INTD_2_HIGH_29	Navigator second hi interrupt
259	QMSS_INTD_2_HIGH_30	Navigator second hi interrupt
260	QMSS_INTD_2_HIGH_31	Navigator second hi interrupt
261	QMSS_INTD_2_LOW_0	Navigator second interrupt
262	QMSS_INTD_2_LOW_1	Navigator second interrupt
263	QMSS_INTD_2_LOW_2	Navigator second interrupt
264	QMSS_INTD_2_LOW_3	Navigator second interrupt
265	QMSS_INTD_2_LOW_4	Navigator second interrupt
266	QMSS_INTD_2_LOW_5	Navigator second interrupt
267	QMSS_INTD_2_LOW_6	Navigator second interrupt
268	QMSS_INTD_2_LOW_7	Navigator second interrupt
269	QMSS_INTD_2_LOW_8	Navigator second interrupt
270	QMSS_INTD_2_LOW_9	Navigator second interrupt
271	QMSS_INTD_2_LOW_10	Navigator second interrupt
272	QMSS_INTD_2_LOW_11	Navigator second interrupt
273	QMSS_INTD_2_LOW_12	Navigator second interrupt
274	QMSS_INTD_2_LOW_13	Navigator second interrupt
275	QMSS_INTD_2_LOW_14	Navigator second interrupt
276	QMSS_INTD_2_LOW_15	Navigator second interrupt
277	UART_0_UARTINT	UART0 interrupt
278	UART_0_URXEVT	UART0 receive event
279	UART_0_UTXEVT	UART0 transmit event
280	UART_1_UARTINT	UART1 interrupt
281	UART_1_URXEVT	UART1 receive event
282	UART_1_UTXEVT	UART1 transmit event
283	I2C_0_INT	I2C interrupt
284	I2C_0_REVT	I2C receive event
285	I2C_0_XEVT	I2C transmit event
286	I2C_1_INT	I2C interrupt
287	I2C_1_REVT	I2C receive event
288	I2C_1_XEVT	I2C transmit event
289	I2C_2_INT	I2C interrupt
290	I2C_2_REVT	I2C receive event
291	I2C_2_XEVT	I2C transmit event
292	SPI_0_INT0	SPI interrupt
293	SPI_0_INT1	SPI interrupt
294	SPI_0_XEVT	SPI DMA TX event
295	SPI_0_REVT	SPI DMA RX event
296	SPI_1_INT0	SPI interrupt
297	SPI_1_INT1	SPI interrupt
298	SPI_1_XEVT	SPI DMA TX event
299	SPI_1_REVT	SPI DMA RX event
300	SPI_2_INT0	SPI interrupt
301	SPI_2_INT1	SPI interrupt

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
302	SPI_2_XEVT	SPI DMA TX event
303	SPI_2_REVT	SPI DMA RX event
304	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
305	DBGTBR_ACQCOMP	Debug Trace buffer (TBR) acquisition has been completed
306	ARM_TBR_DMA	ARM Trace Buffer (TBR) DMA event
307	ARM_TBR_ACQ	ARM Trace Buffer (TBR) Acquisition has been completed
308	NETCP_MDIO_LINK_INT0	Packet Accelerator 1subsystem MDIO interrupt
309	NETCP_MDIO_LINK_INT1	Packet Accelerator 1subsystem MDIO interrupt
310	NETCP_MDIO_USER_INT0	Packet Accelerator 1subsystem MDIO interrupt
311	NETCP_MDIO_USER_INT1	Packet Accelerator 1subsystem MDIO interrupt
312	NETCP_MISC_INT	Packet Accelerator 1subsystem MDIO interrupt
313	Reserved	
314	EDMACC_0_GINT	EDMA3CC0 global completion interrupt
315	EDMACC_0_TC_0_INT	EDMA3CC0 individual completion interrupt
316	EDMACC_0_TC_1_INT	EDMA3CC0 individual completion interrupt
317	EDMACC_0_TC_2_INT	EDMA3CC0 individual completion interrupt
318	EDMACC_0_TC_3_INT	EDMA3CC0 individual completion interrupt
319	EDMACC_0_TC_4_INT	EDMA3CC0 individual completion interrupt
320	EDMACC_0_TC_5_INT	EDMA3CC0 individual completion interrupt
321	EDMACC_0_TC_6_INT	EDMA3CC0 individual completion interrupt
322	EDMACC_0_TC_7_INT	EDMA3CC0 individual completion interrupt
323	EDMACC_1_GINT	EDMA3CC1 global completion interrupt
324	EDMACC_1_TC_0_INT	EDMA3CC1 individual completion interrupt
325	EDMACC_1_TC_1_INT	EDMA3CC1 individual completion interrupt
326	EDMACC_1_TC_2_INT	EDMA3CC1 individual completion interrupt
327	EDMACC_1_TC_3_INT	EDMA3CC1 individual completion interrupt
328	EDMACC_1_TC_4_INT	EDMA3CC1 individual completion interrupt
329	EDMACC_1_TC_5_INT	EDMA3CC1 individual completion interrupt
330	EDMACC_1_TC_6_INT	EDMA3CC1 individual completion interrupt
331	EDMACC_1_TC_7_INT	EDMA3CC1 individual completion interrupt
332	EDMACC_2_GINT	EDMA3CC2 global completion interrupt
333	EDMACC_2_TC_0_INT	EDMA3CC2 individual completion interrupt
334	EDMACC_2_TC_1_INT	EDMA3CC2 individual completion interrupt
335	EDMACC_2_TC_2_INT	EDMA3CC2 individual completion interrupt
336	EDMACC_2_TC_3_INT	EDMA3CC2 individual completion interrupt
337	EDMACC_2_TC_4_INT	EDMA3CC2 individual completion interrupt
338	EDMACC_2_TC_5_INT	EDMA3CC2 individual completion interrupt
339	EDMACC_2_TC_6_INT	EDMA3CC2 individual completion interrupt
340	EDMACC_2_TC_7_INT	EDMA3CC2 individual completion interrupt
341	EDMACC_3_GINT	EDMA3CC3 global completion interrupt
342	EDMACC_3_TC_0_INT	EDMA3CC3 individual completion interrupt
343	EDMACC_3_TC_1_INT	EDMA3CC3 individual completion interrupt
344	EDMACC_3_TC_2_INT	EDMA3CC3 individual completion interrupt
345	EDMACC_3_TC_3_INT	EDMA3CC3 individual completion interrupt
346	EDMACC_3_TC_4_INT	EDMA3CC3 individual completion interrupt
347	EDMACC_3_TC_5_INT	EDMA3CC3 individual completion interrupt
348	EDMACC_3_TC_6_INT	EDMA3CC3 individual completion interrupt

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
349	EDMACC_3_TC_7_INT	EDMA3CC3 individual completion interrupt
350	EDMACC_4_GINT	EDMA3CC4 global completion interrupt
351	EDMACC_4_TC_0_INT	EDMA3CC4 individual completion interrupt
352	EDMACC_4_TC_1_INT	EDMA3CC4 individual completion interrupt
353	EDMACC_4_TC_2_INT	EDMA3CC4 individual completion interrupt
354	EDMACC_4_TC_3_INT	EDMA3CC4 individual completion interrupt
355	EDMACC_4_TC_4_INT	EDMA3CC4 individual completion interrupt
356	EDMACC_4_TC_5_INT	EDMA3CC4 individual completion interrupt
357	EDMACC_4_TC_6_INT	EDMA3CC4 individual completion interrupt
358	EDMACC_4_TC_7_INT	EDMA3CC4 individual completion interrupt
359	SR_0_PO_VCON_SMPSERR_INT	SmartReflex SMPS error interrupt
360	SR_0_SMARTREFLEX_INTREQ0	SmartReflex controller interrupt
361	SR_0_SMARTREFLEX_INTREQ1	SmartReflex controller interrupt
362	SR_0_SMARTREFLEX_INTREQ2	SmartReflex controller interrupt
363	SR_0_SMARTREFLEX_INTREQ3	SmartReflex controller interrupt
364	SR_0_VPNOSMPSACK	SmartReflex VPVOLTUPDATE has been asserted, but SMPS has not been responded to in a defined time interval
365	SR_0_VPEQVALUE	SmartReflex SRSINTERUPT is asserted, but the new voltage is not different from the current SMPS voltage
366	SR_0_VPMAXVDD	SmartReflex. The new voltage required is equal to or greater than MaxVdd
367	SR_0_VPMINVDD	SmartReflex. The new voltage required is equal to or less than MinVdd
368	SR_0_VPINIDLE	SmartReflex indicating that the FSM of voltage processor is in idle
369	SR_0_VPOPPCHANGEDONE	SmartReflex indicating that the average frequency error is within the desired limit
370	SR_0_VPSMPSACK	SmartReflex VPVOLTUPDATE asserted and SMPS has acknowledged in a defined time interval
371	SR_0_SR_TEMPSENSOR	SmartReflex temperature threshold crossing interrupt
372	SR_0_SR_TIMERINT	SmartReflex internal timer expiration interrupt
373	PCIE_1_INT0	PCIE1 legacy INTA interrupt
374	PCIE_1_INT1	PCIE1 legacy INTB interrupt
375	PCIE_1_INT2	PCIE1 legacy INTC interrupt
376	PCIE_1_INT3	PCIE1 legacy INTD interrupt
377	PCIE_1_INT4	PCIE1 MSI interrupt
378	PCIE_1_INT5	PCIE1 MSI interrupt
379	PCIE_1_INT6	PCIE1 MSI interrupt
380	PCIE_1_INT7	PCIE1 MSI interrupt
381	PCIE_1_INT8	PCIE1 MSI interrupt
382	PCIE_1_INT9	PCIE1 MSI interrupt
383	PCIE_1_INT10	PCIE1 MSI interrupt
384	PCIE_1_INT11	PCIE1 MSI interrupt
385	PCIE_1_INT12	PCIE1 error interrupt
386	PCIE_1_INT13	PCIE1 power management interrupt
387	HYPERLINK_0_INT	HyperLink interrupt
388	DDR3_ERR	DDR3 interrupt
389	ARM_NCTIIRQ0	ARM cross trigger (CTI) IRQ interrupt
390	ARM_NCTIIRQ1	ARM cross trigger (CTI) IRQ interrupt
391	ARM_NCTIIRQ2	ARM cross trigger (CTI) IRQ interrupt
392	ARM_NCTIIRQ3	ARM cross trigger (CTI) IRQ interrupt

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
393	Reserved	Reserved
394	Reserved	Reserved
395	Reserved	Reserved
396	Reserved	Reserved
397	Reserved	Reserved
398	Reserved	Reserved
399	Reserved	Reserved
400	Reserved	Reserved
401	Reserved	Reserved
402	10GbE_LINK_INT0	10 Gigabit Ethernet subsystem MDIO interrupt
403	10GbE_USER_INT0	10 Gigabit Ethernet subsystem MDIO interrupt
404	10GbE_LINK_INT1	10 Gigabit Ethernet subsystem MDIO interrupt
405	10GbE_USER_INT1	10 Gigabit Ethernet subsystem MDIO interrupt
406	10GbE_MISC_INT	10 Gigabit Ethernet subsystem MDIO interrupt
407	10GbE_INT_PKTDMA_0	10 Gigabit Ethernet Packet DMA starvation interrupt
408	Reserved	
409	Reserved	
410	Reserved	
411	Reserved	
412	Reserved	
413	Reserved	
414	USB_1_INT00	USB 1 event ring 0 interrupt
415	USB_1_INT01	USB 1 event ring 1 interrupt
416	USB_1_INT02	USB 1 event ring 2 interrupt
417	USB_1_INT03	USB 1 event ring 3 interrupt
418	USB_1_INT04	USB 1 event ring 4 interrupt
419	USB_1_INT05	USB 1 event ring 5 interrupt
420	USB_1_INT06	USB 1 event ring 6 interrupt
421	USB_1_INT07	USB 1 event ring 7 interrupt
422	USB_1_INT08	USB 1 event ring 8 interrupt
423	USB_1_INT09	USB 1 event ring 9 interrupt
424	USB_1_INT10	USB 1 event ring 10 interrupt
425	USB_1_INT11	USB 1 event ring 11 interrupt
426	USB_1_INT12	USB 1 event ring 12 interrupt
427	USB_1_INT13	USB 1 event ring 13 interrupt
428	USB_1_INT14	USB 1 event ring 14 interrupt
429	USB_1_INT15	USB 1 event ring 15 interrupt
430	USB_1_OABSINT	USB 1 OABS interrupt
431	USB_1_MISCINT	USB 1 miscellaneous interrupt
432	NETCP_GLOBAL_STARVE	NETCP GLOBAL interrupt
433	NETCP_LOCAL_STARVE	NETCP LOCAL interrupt
434	NETCP_PA_ECC_INT	NETCP PA ECC interrupt
435	NETCP_SA_ECC_INT	NETCP SA ECC interrupt
436	NETCP_SWITCH_ECC_INT	NETCP SWITCH ECC interrupt
437	NETCP_SWITCH_STAT_INT0	NETCP SWITCH STAT interrupt
438	NETCP_SWITCH_STAT_INT1	NETCP SWITCH STAT interrupt
439	NETCP_SWITCH_STAT_INT2	NETCP SWITCH STAT interrupt

Table 6-22. System Event Mapping — ARM CorePac Interrupts (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
440	NETCP_SWITCH_STAT_INT3	NETCP SWITCH STAT interrupt
441	NETCP_SWITCH_STAT_INT4	NETCP SWITCH STAT interrupt
442	NETCP_SWITCH_STAT_INT5	NETCP SWITCH STAT interrupt
443	NETCP_SWITCH_STAT_INT6	NETCP SWITCH STAT interrupt
444	NETCP_SWITCH_STAT_INT7	NETCP SWITCH STAT interrupt
445	NETCP_SWITCH_INT	NETCP SWITCH interrupt
446	NETCP_SWITCH_STAT_INT0	NETCP SWITCH STAT interrupt
447	Reserved	Reserved
448	CIC_2_OUT29	CIC2 interrupt
449	CIC_2_OUT30	CIC2 interrupt
450	CIC_2_OUT31	CIC2 interrupt
451	CIC_2_OUT32	CIC2 interrupt
452	CIC_2_OUT33	CIC2 interrupt
453	CIC_2_OUT34	CIC2 interrupt
454	CIC_2_OUT35	CIC2 interrupt
455	CIC_2_OUT36	CIC2 interrupt
456	CIC_2_OUT37	CIC2 interrupt
457	CIC_2_OUT38	CIC2 interrupt
458	CIC_2_OUT39	CIC2 interrupt
459	CIC_2_OUT40	CIC2 interrupt
460	CIC_2_OUT41	CIC2 interrupt
461	CIC_2_OUT42	CIC2 interrupt
462	CIC_2_OUT43	CIC2 interrupt
463	CIC_2_OUT44	CIC2 interrupt
464	CIC_2_OUT45	CIC2 interrupt
465	CIC_2_OUT46	CIC2 interrupt
466	CIC_2_OUT47	CIC2 interrupt
467	CIC_2_OUT18	CIC2 interrupt
468	CIC_2_OUT19	CIC2 interrupt
469	CIC_2_OUT22	CIC2 interrupt
470	CIC_2_OUT23	CIC2 interrupt
471	CIC_2_OUT50	CIC2 interrupt
472	CIC_2_OUT51	CIC2 interrupt
473	CIC_2_OUT66	CIC2 interrupt
474	CIC_2_OUT67	CIC2 interrupt
475	CIC_2_OUT88	CIC2 interrupt
476	CIC_2_OUT89	CIC2 interrupt
477	CIC_2_OUT90	CIC2 interrupt
478	CIC_2_OUT91	CIC2 interrupt
479	CIC_2_OUT92	CIC2 interrupt

Table 6-23 lists the CIC2 event inputs.

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink)

EVENT NO.	EVENT NAME	DESCRIPTION
0	GPIO_INT8	GPIO interrupt
1	GPIO_INT9	GPIO interrupt
2	GPIO_INT10	GPIO interrupt

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
3	GPIO_INT11	GPIO interrupt
4	GPIO_INT12	GPIO interrupt
5	GPIO_INT13	GPIO interrupt
6	GPIO_INT14	GPIO interrupt
7	GPIO_INT15	GPIO interrupt
8	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
9	Reserved	Reserved
10	Reserved	Reserved
11	Reserved	Reserved
12	Reserved	Reserved
13	Reserved	Reserved
14	Reserved	Reserved
15	Reserved	Reserved
16	Reserved	Reserved
17	Reserved	Reserved
18	Reserved	Reserved
19	Reserved	Reserved
20	Reserved	Reserved
21	Reserved	Reserved
22	Reserved	Reserved
23	DFT_PBIST_CPU_INT	Reserved
24	QMSS_INTD_1_HIGH_16	Navigator interrupt
25	QMSS_INTD_1_HIGH_17	Navigator interrupt
26	QMSS_INTD_1_HIGH_18	Navigator interrupt
27	QMSS_INTD_1_HIGH_19	Navigator interrupt
28	QMSS_INTD_1_HIGH_20	Navigator interrupt
29	QMSS_INTD_1_HIGH_21	Navigator interrupt
30	QMSS_INTD_1_HIGH_22	Navigator interrupt
31	QMSS_INTD_1_HIGH_23	Navigator interrupt
32	QMSS_INTD_1_HIGH_24	Navigator interrupt
33	QMSS_INTD_1_HIGH_25	Navigator interrupt
34	QMSS_INTD_1_HIGH_26	Navigator interrupt
35	QMSS_INTD_1_HIGH_27	Navigator interrupt
36	QMSS_INTD_1_HIGH_28	Navigator interrupt
37	QMSS_INTD_1_HIGH_29	Navigator interrupt
38	QMSS_INTD_1_HIGH_30	Navigator interrupt
39	QMSS_INTD_1_HIGH_31	Navigator interrupt
40	Reserved	Reserved
41	Reserved	Reserved
42	Reserved	Reserved
43	Reserved	Reserved
44	Reserved	Reserved
45	Reserved	Reserved
46	Reserved	Reserved
47	Reserved	Reserved
48	Reserved	Reserved
49	TRACER_DDR_INT	Tracer sliding time window interrupt for MSMC-DDR3

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
50	TRACER_MSMC_0_INT	Tracer sliding time window interrupt for MSMC SRAM bank0
51	TRACER_MSMC_1_INT	Tracer sliding time window interrupt for MSMC SRAM bank1
52	TRACER_MSMC_2_INT	Tracer sliding time window interrupt for MSMC SRAM bank2
53	TRACER_MSMC_3_INT	Tracer sliding time window interrupt for MSMC SRAM bank3
54	TRACER_CFG_INT	Tracer sliding time window interrupt for TeraNet CFG
55	TRACER_QMSS_QM_CFG1_INT	Tracer sliding time window interrupt for Navigator CFG1 slave port
56	TRACER_QMSS_DMA_INT	Tracer sliding time window interrupt for Navigator VBUSM slave port
57	TRACER_SEM_INT	Tracer sliding time window interrupt for Semaphore interrupt
58	Reserved	Reserved
59	Reserved	Reserved
60	Reserved	Reserved
61	Reserved	Reserved
62	BOOTCFG_INT	BOOTCFG error interrupt
63	NETCP_0_PKTDMA_INT0	Packet Accelerator0 Packet DMA starvation interrupt
64	MPU_0_INT	MPU0 interrupt
65	MSMC_SCRUB_CERROR	MSMC error interrupt
66	MPU_1_INT	MPU1 interrupt
67	Reserved	Reserved
68	MPU_2_INT	MPU2 interrupt
69	QMSS_INTD_1_PKTDMA_0	Navigator Packet DMA interrupt
70	Reserved	Reserved
71	QMSS_INTD_1_PKTDMA_1	Navigator Packet DMA interrupt
72	MSMC_DEDC_CERROR	MSMC error interrupt
73	MSMC_DEDC_NC_ERROR	MSMC error interrupt
74	MSMC_SCRUB_NC_ERROR	MSMC error interrupt
75	Reserved	Reserved
76	MSMC_MPF_ERROR0	Memory protection fault indicators for system master PrivID = 0
77	Reserved	Reserved
78	Reserved	Reserved
79	Reserved	Reserved
80	Reserved	Reserved
81	Reserved	Reserved
82	Reserved	Reserved
83	Reserved	Reserved
84	MSMC_MPF_ERROR8	Memory protection fault indicators for system master PrivID = 8
85	MSMC_MPF_ERROR9	Memory protection fault indicators for system master PrivID = 9
86	MSMC_MPF_ERROR10	Memory protection fault indicators for system master PrivID = 10
87	MSMC_MPF_ERROR11	Memory protection fault indicators for system master PrivID = 11
88	MSMC_MPF_ERROR12	Memory protection fault indicators for system master PrivID = 12
89	MSMC_MPF_ERROR13	Memory protection fault indicators for system master PrivID = 13
90	MSMC_MPF_ERROR14	Memory protection fault indicators for system master PrivID = 14
91	MSMC_MPF_ERROR15	Memory protection fault indicators for system master PrivID = 15
92	Reserved	Reserved
93	GPIO_INT16	GPIO interrupt
94	GPIO_INT17	GPIO interrupt
95	GPIO_INT18	GPIO interrupt
96	GPIO_INT19	GPIO interrupt

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
97	GPIO_INT20	GPIO interrupt
98	GPIO_INT21	GPIO interrupt
99	GPIO_INT22	GPIO interrupt
100	GPIO_INT23	GPIO interrupt
101	GPIO_INT24	GPIO interrupt
102	GPIO_INT25	GPIO interrupt
103	GPIO_INT26	GPIO interrupt
104	GPIO_INT27	GPIO interrupt
105	GPIO_INT28	GPIO interrupt
106	GPIO_INT29	GPIO interrupt
107	GPIO_INT30	GPIO interrupt
108	GPIO_INT31	GPIO interrupt
109	Reserved	Reserved
110	Reserved	Reserved
111	Reserved	Reserved
112	Reserved	Reserved
113	Reserved	Reserved
114	Reserved	Reserved
115	Reserved	Reserved
116	Reserved	Reserved
117	AEMIF_EASYNCERR	Asynchronous EMIF16 error interrupt
118	Reserved	Reserved
119	Reserved	Reserved
120	Reserved	Reserved
121	Reserved	Reserved
122	Reserved	Reserved
123	Reserved	Reserved
124	Reserved	Reserved
125	Reserved	Reserved
126	Reserved	Reserved
127	Reserved	Reserved
128	Reserved	Reserved
129	Reserved	Reserved
130	Reserved	Reserved
131	Reserved	Reserved
132	Reserved	Reserved
133	Reserved	Reserved
134	Reserved	Reserved
135	Reserved	Reserved
136	Reserved	Reserved
137	Reserved	Reserved
138	QMSS_INTD_1_HIGH_0	Navigator hi interrupt
139	QMSS_INTD_1_HIGH_1	Navigator hi interrupt
140	QMSS_INTD_1_HIGH_2	Navigator hi interrupt
141	QMSS_INTD_1_HIGH_3	Navigator hi interrupt
142	QMSS_INTD_1_HIGH_4	Navigator hi interrupt
143	QMSS_INTD_1_HIGH_5	Navigator hi interrupt

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
144	QMSS_INTD_1_HIGH_6	Navigator hi interrupt
145	QMSS_INTD_1_HIGH_7	Navigator hi interrupt
146	QMSS_INTD_1_HIGH_8	Navigator hi interrupt
147	QMSS_INTD_1_HIGH_9	Navigator hi interrupt
148	QMSS_INTD_1_HIGH_10	Navigator hi interrupt
149	QMSS_INTD_1_HIGH_11	Navigator hi interrupt
150	QMSS_INTD_1_HIGH_12	Navigator hi interrupt
151	QMSS_INTD_1_HIGH_13	Navigator hi interrupt
152	QMSS_INTD_1_HIGH_14	Navigator hi interrupt
153	QMSS_INTD_1_HIGH_15	Navigator hi interrupt
154	QMSS_INTD_2_HIGH_0	Navigator second hi interrupt
155	QMSS_INTD_2_HIGH_1	Navigator second hi interrupt
156	QMSS_INTD_2_HIGH_2	Navigator second hi interrupt
157	QMSS_INTD_2_HIGH_3	Navigator second hi interrupt
158	QMSS_INTD_2_HIGH_4	Navigator second hi interrupt
159	QMSS_INTD_2_HIGH_5	Navigator second hi interrupt
160	QMSS_INTD_2_HIGH_6	Navigator second hi interrupt
161	QMSS_INTD_2_HIGH_7	Navigator second hi interrupt
162	QMSS_INTD_2_HIGH_8	Navigator second hi interrupt
163	QMSS_INTD_2_HIGH_9	Navigator second hi interrupt
164	QMSS_INTD_2_HIGH_10	Navigator second hi interrupt
165	QMSS_INTD_2_HIGH_11	Navigator second hi interrupt
166	QMSS_INTD_2_HIGH_12	Navigator second hi interrupt
167	QMSS_INTD_2_HIGH_13	Navigator second hi interrupt
168	QMSS_INTD_2_HIGH_14	Navigator second hi interrupt
169	QMSS_INTD_2_HIGH_15	Navigator second hi interrupt
170	QMSS_INTD_2_HIGH_16	Navigator second hi interrupt
171	QMSS_INTD_2_HIGH_17	Navigator second hi interrupt
172	QMSS_INTD_2_HIGH_18	Navigator second hi interrupt
173	QMSS_INTD_2_HIGH_19	Navigator second hi interrupt
174	QMSS_INTD_2_HIGH_20	Navigator second hi interrupt
175	QMSS_INTD_2_HIGH_21	Navigator second hi interrupt
176	QMSS_INTD_2_HIGH_22	Navigator second hi interrupt
177	QMSS_INTD_2_HIGH_23	Navigator second hi interrupt
178	QMSS_INTD_2_HIGH_24	Navigator second hi interrupt
179	QMSS_INTD_2_HIGH_25	Navigator second hi interrupt
180	QMSS_INTD_2_HIGH_26	Navigator second hi interrupt
181	QMSS_INTD_2_HIGH_27	Navigator second hi interrupt
182	QMSS_INTD_2_HIGH_28	Navigator second hi interrupt
183	QMSS_INTD_2_HIGH_29	Navigator second hi interrupt
184	QMSS_INTD_2_HIGH_30	Navigator second hi interrupt
185	QMSS_INTD_2_HIGH_31	Navigator second hi interrupt
186	MPU_12_INT	MPU12 addressing violation interrupt and protection violation interrupt
187	MPU_13_INT	MPU13 addressing violation interrupt and protection violation interrupt
188	MPU_14_INT	MPU14 addressing violation interrupt and protection violation interrupt
189	MPU_15_INT	MPU15 addressing violation interrupt and protection violation interrupt
190	Reserved	Reserved

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
191	Reserved	Reserved
192	Reserved	Reserved
193	Reserved	Reserved
194	Reserved	Reserved
195	Reserved	Reserved
196	Reserved	Reserved
197	Reserved	Reserved
198	Reserved	Reserved
199	TRACER_QMSS_QM_CFG2_INT	Tracer sliding time window interrupt for Navigator CFG2 slave port
200	TRACER_EDMACC_0	Tracer sliding time window interrupt foR EDMA3CC0
201	TRACER_EDMACC_123_INT	Tracer sliding time window interrupt for EDMA3CC1, EDMA3CC2, and EDMA3CC3
202	TRACER_CIC_INT	Tracer sliding time window interrupt for interrupt controllers (CIC)
203	Reserved	Reserved
204	MPU_5_INT	MPU5 addressing violation interrupt and protection violation interrupt
205	Reserved	Reserved
206	MPU_7_INT	MPU7 addressing violation interrupt and protection violation interrupt
207	MPU_8_INT	MPU8 addressing violation interrupt and protection violation interrupt
208	Reserved	Reserved
209	Reserved	Reserved
210	Reserved	Reserved
211	DDR3_0_ERR	DDR3 error interrupt
212	HYPERLINK_0_INT	HyperLink interrupt
213	EDMACC_0_ERRINT	EDMA3CC0 error interrupt
214	EDMACC_0_MPINT	EDMA3CC0 memory protection interrupt
215	EDMACC_0_TC_0_ERRINT	EDMA3CC0 TPTC0 error interrupt
216	EDMACC_0_TC_1_ERRINT	EDMA3CC0 TPTC1 error interrupt
217	EDMACC_1_ERRINT	EDMA3CC1 error interrupt
218	EDMACC_1_MPINT	EDMA3CC1 memory protection interrupt
219	EDMACC_1_TC_0_ERRINT	EDMA3CC1 TPTC0 error interrupt
220	EDMACC_1_TC_1_ERRINT	EDMA3CC1 TPTC1 error interrupt
221	EDMACC_1_TC_2_ERRINT	EDMA3CC1 TPTC2 error interrupt
222	EDMACC_1_TC_3_ERRINT	EDMA3CC1 TPTC3 error interrupt
223	EDMACC_2_ERRINT	EDMA3CC2 error interrupt
224	EDMACC_2_MPINT	EDMA3CC2 memory protection interrupt
225	EDMACC_2_TC_0_ERRINT	EDMA3CC2 TPTC0 error interrupt
226	EDMACC_2_TC_1_ERRINT	EDMA3CC2 TPTC1 error interrupt
227	EDMACC_2_TC_2_ERRINT	EDMA3CC2 TPTC2 error interrupt
228	EDMACC_2_TC_3_ERRINT	EDMA3CC2 TPTC3 error interrupt
229	EDMACC_3_ERRINT	EDMA3CC3 error interrupt
230	EDMACC_3_MPINT	EDMA3CC3 memory protection interrupt
231	EDMACC_3_TC_0_ERRINT	EDMA3CC3 TPTC0 error interrupt
232	EDMACC_3_TC_1_ERRINT	EDMA3CC3 TPTC1 error interrupt
233	EDMACC_4_ERRINT	EDMA3CC4 error interrupt
234	EDMACC_4_MPINT	EDMA3CC4 memory protection interrupt
235	EDMACC_4_TC_0_ERRINT	EDMA3CC4 TPTC0 error interrupt
236	EDMACC_4_TC_1_ERRINT	EDMA3CC4 TPTC1 error interrupt
237	QMSS_QUE_PEND_652	Navigator transmit queue pending event for indicated queue

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
238	QMSS_QUE_PEND_653	Navigator transmit queue pending event for indicated queue
239	QMSS_QUE_PEND_654	Navigator transmit queue pending event for indicated queue
240	QMSS_QUE_PEND_655	Navigator transmit queue pending event for indicated queue
241	QMSS_QUE_PEND_656	Navigator transmit queue pending event for indicated queue
242	QMSS_QUE_PEND_657	Navigator transmit queue pending event for indicated queue
243	QMSS_QUE_PEND_658	Navigator transmit queue pending event for indicated queue
244	QMSS_QUE_PEND_659	Navigator transmit queue pending event for indicated queue
245	QMSS_QUE_PEND_660	Navigator transmit queue pending event for indicated queue
246	QMSS_QUE_PEND_661	Navigator transmit queue pending event for indicated queue
247	QMSS_QUE_PEND_662	Navigator transmit queue pending event for indicated queue
248	QMSS_QUE_PEND_663	Navigator transmit queue pending event for indicated queue
249	QMSS_QUE_PEND_664	Navigator transmit queue pending event for indicated queue
250	QMSS_QUE_PEND_665	Navigator transmit queue pending event for indicated queue
251	QMSS_QUE_PEND_666	Navigator transmit queue pending event for indicated queue
252	QMSS_QUE_PEND_667	Navigator transmit queue pending event for indicated queue
253	QMSS_QUE_PEND_668	Navigator transmit queue pending event for indicated queue
254	QMSS_QUE_PEND_669	Navigator transmit queue pending event for indicated queue
255	QMSS_QUE_PEND_670	Navigator transmit queue pending event for indicated queue
256	QMSS_QUE_PEND_671	Navigator transmit queue pending event for indicated queue
257	QMSS_QUE_PEND_672	Navigator transmit queue pending event for indicated queue
258	QMSS_QUE_PEND_673	Navigator transmit queue pending event for indicated queue
259	QMSS_QUE_PEND_674	Navigator transmit queue pending event for indicated queue
260	QMSS_QUE_PEND_675	Navigator transmit queue pending event for indicated queue
261	QMSS_QUE_PEND_676	Navigator transmit queue pending event for indicated queue
262	QMSS_QUE_PEND_677	Navigator transmit queue pending event for indicated queue
263	QMSS_QUE_PEND_678	Navigator transmit queue pending event for indicated queue
264	QMSS_QUE_PEND_679	Navigator transmit queue pending event for indicated queue
265	QMSS_QUE_PEND_680	Navigator transmit queue pending event for indicated queue
266	QMSS_QUE_PEND_681	Navigator transmit queue pending event for indicated queue
267	QMSS_QUE_PEND_682	Navigator transmit queue pending event for indicated queue
268	QMSS_QUE_PEND_683	Navigator transmit queue pending event for indicated queue
269	QMSS_QUE_PEND_684	Navigator transmit queue pending event for indicated queue
270	QMSS_QUE_PEND_685	Navigator transmit queue pending event for indicated queue
271	QMSS_QUE_PEND_686	Navigator transmit queue pending event for indicated queue
272	QMSS_QUE_PEND_687	Navigator transmit queue pending event for indicated queue
273	QMSS_QUE_PEND_688	Navigator transmit queue pending event for indicated queue
274	QMSS_QUE_PEND_689	Navigator transmit queue pending event for indicated queue
275	QMSS_QUE_PEND_690	Navigator transmit queue pending event for indicated queue
276	QMSS_QUE_PEND_691	Navigator transmit queue pending event for indicated queue
277	10GbE_LINK_INT0	10 Gigabit Ethernet subsystem MDIO interrupt
278	10GbE_LINK_INT1	10 Gigabit Ethernet subsystem MDIO interrupt
279	10GbE_USER_INT0	10 Gigabit Ethernet subsystem MDIO interrupt
280	10GbE_USER_INT1	10 Gigabit Ethernet subsystem MDIO interrupt
281	10GbE_MISC_INT	10 Gigabit Ethernet subsystem MDIO interrupt
282	10GbE_INT_PKTDMA_0	10 Gigabit Ethernet Packet DMA starvation interrupt
283	Reserved	Reserved
284	Reserved	Reserved

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
285	Reserved	Reserved
286	Reserved	Reserved
287	Reserved	Reserved
288	Reserved	Reserved
289	Reserved	Reserved
290	Reserved	Reserved
291	SEM_INT8	Semaphore interrupt
292	SEM_INT9	Semaphore interrupt
293	SEM_INT10	Semaphore interrupt
294	SEM_INT11	Semaphore interrupt
295	SEM_INT12	Semaphore interrupt
296	Reserved	Reserved
297	Reserved	Reserved
298	Reserved	Reserved
299	SEM_ERR8	Semaphore error interrupt
300	SEM_ERR9	Semaphore error interrupt
301	SEM_ERR10	Semaphore error interrupt
302	SEM_ERR11	Semaphore error interrupt
303	SEM_ERR12	Semaphore error interrupt
304	QMSS1_ECC_INTR	Navigator ECC error interrupt
305	QMSS_INTD_1_LOW_0	Navigator interrupt
306	QMSS_INTD_1_LOW_1	Navigator interrupt
307	QMSS_INTD_1_LOW_2	Navigator interrupt
308	QMSS_INTD_1_LOW_3	Navigator interrupt
309	QMSS_INTD_1_LOW_4	Navigator interrupt
310	QMSS_INTD_1_LOW_5	Navigator interrupt
311	QMSS_INTD_1_LOW_6	Navigator interrupt
312	QMSS_INTD_1_LOW_7	Navigator interrupt
313	QMSS_INTD_1_LOW_8	Navigator interrupt
314	QMSS_INTD_1_LOW_9	Navigator interrupt
315	QMSS_INTD_1_LOW_10	Navigator interrupt
316	QMSS_INTD_1_LOW_11	Navigator interrupt
317	QMSS_INTD_1_LOW_12	Navigator interrupt
318	QMSS_INTD_1_LOW_13	Navigator interrupt
319	QMSS_INTD_1_LOW_14	Navigator interrupt
320	QMSS_INTD_1_LOW_15	Navigator interrupt
321	QMSS_INTD_2_LOW_0	Navigator second interrupt
322	QMSS_INTD_2_LOW_1	Navigator second interrupt
323	QMSS_INTD_2_LOW_2	Navigator second interrupt
324	QMSS_INTD_2_LOW_3	Navigator second interrupt
325	QMSS_INTD_2_LOW_4	Navigator second interrupt
326	QMSS_INTD_2_LOW_5	Navigator second interrupt
327	QMSS_INTD_2_LOW_6	Navigator second interrupt
328	QMSS_INTD_2_LOW_7	Navigator second interrupt
329	QMSS_INTD_2_LOW_8	Navigator second interrupt
330	QMSS_INTD_2_LOW_9	Navigator second interrupt
331	QMSS_INTD_2_LOW_10	Navigator second interrupt

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
332	QMSS_INTD_2_LOW_11	Navigator second interrupt
333	QMSS_INTD_2_LOW_12	Navigator second interrupt
334	QMSS_INTD_2_LOW_13	Navigator second interrupt
335	QMSS_INTD_2_LOW_14	Navigator second interrupt
336	QMSS_INTD_2_LOW_15	Navigator interrupt
337	NETCP_MDIO_LINK_INT0	Packet Accelerator subsystem MDIO interrupt
338	NETCP_MDIO_LINK_INT1	Packet Accelerator subsystem MDIO interrupt
339	NETCP_MDIO_USER_INT0	Packet Accelerator subsystem MDIO interrupt
340	NETCP_MDIO_USER_INT1	Packet Accelerator subsystem MDIO interrupt
341	NETCP_MISC_INT	Packet Accelerator subsystem MDIO interrupt
342	NETCP_GLOBAL_STARVE_INT	Packet Accelerator interrupt
343	NETCP_LOCAL_STARVE_INT	Packet Accelerator interrupt
344	NETCP_PA_ECC_INT	Packet Accelerator interrupt
345	NETCP_SA_ECC_INT	Packet Accelerator interrupt
346	NETCP_SWITCH_ECC_INT	NETCP SWITCH ECC interrupt
347	NETCP_SWITCH_STAT_INT0	NETCP SWITCH STAT interrupt
348	NETCP_SWITCH_STAT_INT1	NETCP SWITCH STAT interrupt
349	NETCP_SWITCH_STAT_INT2	NETCP SWITCH STAT interrupt
350	NETCP_SWITCH_STAT_INT3	NETCP SWITCH STAT interrupt
351	NETCP_SWITCH_STAT_INT4	NETCP SWITCH STAT interrupt
352	NETCP_SWITCH_STAT_INT5	NETCP SWITCH STAT interrupt
353	NETCP_SWITCH_STAT_INT6	NETCP SWITCH STAT interrupt
354	NETCP_SWITCH_STAT_INT7	NETCP SWITCH STAT interrupt
355	NETCP_SWITCH_STAT_INT8	NETCP SWITCH STAT interrupt
356	NETCP_SWITCH_INT	NETCP SWITCH interrupt
357	Reserved	Reserved
358	Reserved	Reserved
359	Reserved	Reserved
360	Reserved	Reserved
361	Reserved	Reserved
362	PSC_ALLINT	PSC interrupt
363	Reserved	Reserved
364	Reserved	Reserved
365	Reserved	Reserved
366	Reserved	Reserved
367	Reserved	Reserved
368	Reserved	Reserved
369	Reserved	Reserved
370	Reserved	Reserved
371	Reserved	Reserved
372	MPU_9_INT	MPU9 addressing violation interrupt and protection violation interrupt
373	MPU_10_INT	MPU10 addressing violation interrupt and protection violation interrupt
374	MPU_11_INT	MPU11 addressing violation interrupt and protection violation interrupt
375	TRACER_MSMC_4_INT	Tracer sliding time window interrupt for MSMC SRAM bank 4
376	TRACER_MSMC_5_INT	Tracer sliding time window interrupt for MSMC SRAM bank 4
377	TRACER_MSMC_6_INT	Tracer sliding time window interrupt for MSMC SRAM bank 4
378	TRACER_MSMC_7_INT	Tracer sliding time window interrupt for MSMC SRAM bank 4

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
379	TRACER_PCIE1_INT	Tracer sliding time window interrupt for PCIE1
380	Reserved	Reserved
381	Reserved	Reserved
382	Reserved	Reserved
383	Reserved	Reserved
384	TRACER_SPI_ROM_EMIF_INT	Tracer sliding time window interrupt for SPI/ROM/EMIF16 modules
385	Reserved	Reserved
386	TRACER_USB1_INT	Tracer sliding time window interrupt for USB1 CFG port tracer
387	TIMER_8_INTL	Timer interrupt low
388	TIMER_8_INTH	Timer interrupt high
389	TIMER_9_INTL	Timer interrupt low
390	TIMER_9_INTH	Timer interrupt high
391	TIMER_10_INTL	Timer interrupt low
392	TIMER_10_INTH	Timer interrupt high
393	TIMER_11_INTL	Timer interrupt low
394	TIMER_11_INTH	Timer interrupt high
395	TIMER_14_INTL	Timer interrupt low
396	TIMER_14_INTH	Timer interrupt high
397	TIMER_15_INTL	Timer interrupt low
398	TIMER_15_INTH	Timer interrupt high
399	USB_0_INT00	USB 0 event ring 0 interrupt
400	USB_0_INT01	USB 0 event ring 1 interrupt
401	USB_0_INT02	USB 0 event ring 2 interrupt
402	USB_0_INT03	USB 0 event ring 3 interrupt
403	USB_0_INT04	USB 0 event ring 4 interrupt
404	USB_0_INT05	USB 0 event ring 5 interrupt
405	USB_0_INT06	USB 0 event ring 6 interrupt
406	USB_0_INT07	USB 0 event ring 7 interrupt
407	USB_0_INT08	USB 0 event ring 8 interrupt
408	USB_0_INT09	USB 0 event ring 9 interrupt
409	USB_0_INT10	USB 0 event ring 10 interrupt
410	USB_0_INT11	USB 0 event ring 11 interrupt
411	USB_0_INT12	USB 0 event ring 12 interrupt
412	USB_0_INT13	USB 0 event ring 13 interrupt
413	USB_0_INT14	USB 0 event ring 14 interrupt
414	USB_0_INT15	USB 0 event ring 15 interrupt
415	USB_0_MISCINT	USB 0 Miscellaneous interrupt
416	USB_0_OABSINT	USB 0 OABS interrupt
417	Reserved	Reserved
418	USB_1_INT00	USB 1 event ring 0 interrupt
419	USB_1_INT01	USB 1 event ring 1 interrupt
420	USB_1_INT02	USB 1 event ring 2 interrupt
421	USB_1_INT03	USB 1 event ring 3 interrupt
422	USB_1_INT04	USB 1 event ring 4 interrupt
423	USB_1_INT05	USB 1 event ring 5 interrupt
424	USB_1_INT06	USB 1 event ring 6 interrupt
425	USB_1_INT07	USB 1 event ring 7 interrupt

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
426	USB_1_INT08	USB 1 event ring 8 interrupt
427	USB_1_INT09	USB 1 event ring 9 interrupt
428	USB_1_INT10	USB 1 event ring 10 interrupt
429	USB_1_INT11	USB 1 event ring 11 interrupt
430	USB_1_INT12	USB 1 event ring 12 interrupt
431	USB_1_INT13	USB 1 event ring 13 interrupt
432	USB_1_INT14	USB 1 event ring 14 interrupt
433	USB_1_INT15	USB 1 event ring 15 interrupt
434	USB_1_MISCINT	USB 1 miscellaneous interrupt
435	USB_1_OABSINT	USB 1 OABS interrupt
436	Reserved	Reserved
437	Reserved	Reserved
438	Reserved	Reserved
439	Reserved	Reserved
440	Reserved	Reserved
441	Reserved	Reserved
442	Reserved	Reserved
443	Reserved	Reserved
444	Reserved	Reserved
445	Reserved	Reserved
446	TIMER_12_INTL	Timer interrupt low
447	TIMER_12_INTH	Timer interrupt high
448	TIMER_13_INTL	Timer interrupt low
449	TIMER_13_INTH	Timer interrupt high
450	TIMER_16_INTL	Timer interrupt low
451	TIMER_16_INTH	Timer interrupt high
452	TIMER_17_INTL	Timer interrupt low
453	TIMER_17_INTH	Timer interrupt high
454	TIMER_18_INTL	Timer interrupt low
455	TIMER_18_INTH	Timer interrupt high
456	TIMER_19_INTL	Timer interrupt low
457	TIMER_19_INTH	Timer interrupt high
458	Reserved	Reserved
459	RSTMUX_INT8	Boot config watchdog timer expiration event for ARM Core 0
460	RSTMUX_INT9	Boot config watchdog timer expiration event for ARM Core 1
461	RSTMUX_INT10	Boot config watchdog timer expiration event for ARM Core 2
462	RSTMUX_INT11	Boot config watchdog timer expiration event for ARM Core 3
463	GPIO_INT0	GPIO interrupt
464	GPIO_INT1	GPIO interrupt
465	GPIO_INT2	GPIO interrupt
466	GPIO_INT3	GPIO interrupt
467	GPIO_INT4	GPIO interrupt
468	GPIO_INT5	GPIO interrupt
469	GPIO_INT6	GPIO interrupt
470	GPIO_INT7	GPIO interrupt
471	Reserved	Reserved
472	Reserved	Reserved

Table 6-23. CIC2 Event Inputs (Secondary Events for EDMA3CC and Hyperlink) (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
473	Reserved	Reserved
474	Reserved	Reserved
475	Reserved	Reserved
476	Reserved	Reserved
477	Reserved	Reserved
478	Reserved	Reserved

6.3.2 CIC Registers

This section includes the CIC memory map information and registers.

6.3.2.1 CIC2 Register Map

Table 6-24. CIC2 Registers

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
0x0	REVISION_REG	Revision Register
0x10	GLOBAL_ENABLE_HINT_REG	Global Host Int Enable Register
0x20	STATUS_SET_INDEX_REG	Status Set Index Register
0x24	STATUS_CLR_INDEX_REG	Status Clear Index Register
0x28	ENABLE_SET_INDEX_REG	Enable Set Index Register
0x2C	ENABLE_CLR_INDEX_REG	Enable Clear Index Register
0x34	HINT_ENABLE_SET_INDEX_REG	Host Int Enable Set Index Register
0x38	HINT_ENABLE_CLR_INDEX_REG	Host Int Enable Clear Index Register
0x200	RAW_STATUS_REG0	Raw Status Register 0
0x204	RAW_STATUS_REG1	Raw Status Register 1
0x208	RAW_STATUS_REG2	Raw Status Register 2
0x20C	RAW_STATUS_REG3	Raw Status Register 3
0x210	RAW_STATUS_REG4	Raw Status Register 4
0x214	RAW_STATUS_REG5	Raw Status Register 5
0x218	RAW_STATUS_REG6	Raw Status Register 6
0x21C	RAW_STATUS_REG7	Raw Status Register 7
0x220	RAW_STATUS_REG8	Raw Status Register 8
0x224	RAW_STATUS_REG9	Raw Status Register 9
0x228	RAW_STATUS_REG10	Raw Status Register 10
0x22C	RAW_STATUS_REG11	Raw Status Register 11
0x230	RAW_STATUS_REG12	Raw Status Register 12
0x234	RAW_STATUS_REG13	Raw Status Register 13
0x238	RAW_STATUS_REG14	Raw Status Register 14
0x23C	RAW_STATUS_REG15	Raw Status Register 15
0x280	ENA_STATUS_REG0	Enabled Status Register 0
0x284	ENA_STATUS_REG1	Enabled Status Register 1
0x288	ENA_STATUS_REG2	Enabled Status Register 2
0x28C	ENA_STATUS_REG3	Enabled Status Register 3
0x290	ENA_STATUS_REG4	Enabled Status Register 4
0x294	ENA_STATUS_REG5	Enabled Status Register 5
0x298	ENA_STATUS_REG6	Enabled Status Register 6
0x29C	ENA_STATUS_REG7	Enabled Status Register 7
0x2A0	ENA_STATUS_REG8	Enabled Status Register 8
0x2A4	ENA_STATUS_REG9	Enabled Status Register 9
0x2A8	ENA_STATUS_REG10	Enabled Status Register10
0x2AC	ENA_STATUS_REG11	Enabled Status Register 11
0x2B0	ENA_STATUS_REG12	Enabled Status Register 12
0x2B4	ENA_STATUS_REG13	Enabled Status Register 13
0x2B8	ENA_STATUS_REG14	Enabled Status Register 14
0x2BC	ENA_STATUS_REG15	Enabled Status Register 15
0x300	ENABLE_REG0	Enable Register 0
0x304	ENABLE_REG1	Enable Register 1
0x308	ENABLE_REG2	Enable Register 2
0x30C	ENABLE_REG3	Enable Register 3
0x310	ENABLE_REG4	Enable Register 4

Table 6-24. CIC2 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
0x314	ENABLE_REG5	Enable Register 5
0x318	ENABLE_REG6	Enable Register 6
0x31C	ENABLE_REG7	Enable Register 7
0x320	ENABLE_REG8	Enable Register 8
0x324	ENABLE_REG9	Enable Register 9
0x328	ENABLE_REG10	Enable Register 10
0x32C	ENABLE_REG11	Enable Register 11
0x330	ENABLE_REG12	Enable Register 12
0x334	ENABLE_REG13	Enable Register 13
0x338	ENABLE_REG14	Enable Register 14
0x33C	ENABLE_REG15	Enable Register 15
0x380	ENABLE_CLR_REG0	Enable Clear Register 0
0x384	ENABLE_CLR_REG1	Enable Clear Register 1
0x388	ENABLE_CLR_REG2	Enable Clear Register 2
0x38C	ENABLE_CLR_REG3	Enable Clear Register 3
0x390	ENABLE_CLR_REG4	Enable Clear Register 4
0x394	ENABLE_CLR_REG5	Enable Clear Register 5
0x398	ENABLE_CLR_REG6	Enable Clear Register 6
0x39C	ENABLE_CLR_REG7	Enable Clear Register 7
0x3A0	ENABLE_CLR_REG8	Enable Clear Register 8
0x3A4	ENABLE_CLR_REG9	Enable Clear Register 9
0x3A8	ENABLE_CLR_REG10	Enable Clear Register 10
0x3AC	ENABLE_CLR_REG11	Enable Clear Register 11
0x3B0	ENABLE_CLR_REG12	Enable Clear Register 12
0x3B4	ENABLE_CLR_REG13	Enable Clear Register 13
0x3B8	ENABLE_CLR_REG14	Enable Clear Register 14
0x38C	ENABLE_CLR_REG15	Enable Clear Register 15
0x400	CH_MAP_REG0	Interrupt Channel Map Register for 0 to 0+3
0x404	CH_MAP_REG1	Interrupt Channel Map Register for 4 to 4+3
0x408	CH_MAP_REG2	Interrupt Channel Map Register for 8 to 8+3
0x40C	CH_MAP_REG3	Interrupt Channel Map Register for 12 to 12+3
0x410	CH_MAP_REG4	Interrupt Channel Map Register for 16 to 16+3
0x414	CH_MAP_REG5	Interrupt Channel Map Register for 20 to 20+3
0x418	CH_MAP_REG6	Interrupt Channel Map Register for 24 to 24+3
0x41C	CH_MAP_REG7	Interrupt Channel Map Register for 28 to 28+3
0x420	CH_MAP_REG8	Interrupt Channel Map Register for 32 to 32+3
0x424	CH_MAP_REG9	Interrupt Channel Map Register for 36 to 36+3
0x428	CH_MAP_REG10	Interrupt Channel Map Register for 40 to 40+3
0x42C	CH_MAP_REG11	Interrupt Channel Map Register for 44 to 44+3
0x430	CH_MAP_REG12	Interrupt Channel Map Register for 48 to 48+3
0x434	CH_MAP_REG13	Interrupt Channel Map Register for 52 to 52+3
0x438	CH_MAP_REG14	Interrupt Channel Map Register for 56 to 56+3
0x43C	CH_MAP_REG15	Interrupt Channel Map Register for 60 to 60+3
0x5C0	CH_MAP_REG116	Interrupt Channel Map Register for 464 to 464+3
0x5C4	CH_MAP_REG117	Interrupt Channel Map Register for 468 to 468+3
0x5C8	CH_MAP_REG118	Interrupt Channel Map Register for 472 to 472+3
0x5CC	CH_MAP_REG119	Interrupt Channel Map Register for 476 to 476+3

Table 6-24. CIC2 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
0x5D0	CH_MAP_REG120	Interrupt Channel Map Register for 480 to 480+3
0x5D4	CH_MAP_REG121	Interrupt Channel Map Register for 484 to 484+3
0x5D8	CH_MAP_REG122	Interrupt Channel Map Register for 488 to 488+3
0x5DC	CH_MAP_REG123	Interrupt Channel Map Register for 482 to 492+3
0x5E0	CH_MAP_REG124	Interrupt Channel Map Register for 496 to 496+3
0x5E4	CH_MAP_REG125	Interrupt Channel Map Register for 500 to 500+3
0x5E8	CH_MAP_REG126	Interrupt Channel Map Register for 504 to 504+3
0x5EC	CH_MAP_REG127	Interrupt Channel Map Register for 508 to 508+3
0x5F0	CH_MAP_REG128	Interrupt Channel Map Register for 512 to 512+3
0x5F4	CH_MAP_REG129	Interrupt Channel Map Register for 516 to 516+3
0x5F8	CH_MAP_REG130	Interrupt Channel Map Register for 520 to 520+3
0x5FC	CH_MAP_REG131	Interrupt Channel Map Register for 524 to 524+3
0x600	CH_MAP_REG132	Interrupt Channel Map Register for 528 to 528+3
0x604	CH_MAP_REG133	Interrupt Channel Map Register for 532 to 532+3
0x608	CH_MAP_REG134	Interrupt Channel Map Register for 536 to 536+3
0x60C	CH_MAP_REG135	Interrupt Channel Map Register for 540 to 540+3
0x610	CH_MAP_REG136	Interrupt Channel Map Register for 544 to 544+3
0x614	CH_MAP_REG137	Interrupt Channel Map Register for 548 to 548+3
0x618	CH_MAP_REG138	Interrupt Channel Map Register for 552 to 552+3
0x61C	CH_MAP_REG139	Interrupt Channel Map Register for 556 to 556+3
0x620	CH_MAP_REG140	Interrupt Channel Map Register for 560 to 560+3
0x624	CH_MAP_REG141	Interrupt Channel Map Register for 564 to 564+3
0x628	CH_MAP_REG142	Interrupt Channel Map Register for 568 to 568+3
0x62C	CH_MAP_REG143	Interrupt Channel Map Register for 572 to 572+3
0x630	CH_MAP_REG144	Interrupt Channel Map Register for 576 to 576+3
0x634	CH_MAP_REG145	Interrupt Channel Map Register for 580 to 580+3
0x638	CH_MAP_REG146	Interrupt Channel Map Register for 584 to 584+3
0x63C	CH_MAP_REG147	Interrupt Channel Map Register for 588 to 588+3
0x640	CH_MAP_REG148	Interrupt Channel Map Register for 592 to 592+3
0x644	CH_MAP_REG149	Interrupt Channel Map Register for 596 to 596+3
0x648	CH_MAP_REG150	Interrupt Channel Map Register for 600 to 600+3
0x64C	CH_MAP_REG151	Interrupt Channel Map Register for 604 to 604+3
0x650	CH_MAP_REG152	Interrupt Channel Map Register for 608 to 608+3
0x654	CH_MAP_REG153	Interrupt Channel Map Register for 612 to 612+3
0x658	CH_MAP_REG154	Interrupt Channel Map Register for 616 to 616+3
0x65C	CH_MAP_REG155	Interrupt Channel Map Register for 620 to 620+3
0x660	CH_MAP_REG156	Interrupt Channel Map Register for 624 to 624+3
0x664	CH_MAP_REG157	Interrupt Channel Map Register for 628 to 628+3
0x668	CH_MAP_REG158	Interrupt Channel Map Register for 632 to 632+3
0x66C	CH_MAP_REG159	Interrupt Channel Map Register for 636 to 636+3
0x670	CH_MAP_REG160	Interrupt Channel Map Register for 640 to 640+3
0x674	CH_MAP_REG161	Interrupt Channel Map Register for 644 to 644+3
0x678	CH_MAP_REG162	Interrupt Channel Map Register for 648 to 648+3
0x67C	CH_MAP_REG163	Interrupt Channel Map Register for 652 to 652+3
0x680	CH_MAP_REG164	Interrupt Channel Map Register for 656 to 656+3
0x684	CH_MAP_REG165	Interrupt Channel Map Register for 660 to 660+3
0x688	CH_MAP_REG166	Interrupt Channel Map Register for 664 to 664+3

Table 6-24. CIC2 Registers (continued)

ADDRESS OFFSET	REGISTER MNEMONIC	REGISTER NAME
0x68C	CH_MAP_REG167	Interrupt Channel Map Register for 668 to 668+3
0x690	CH_MAP_REG168	Interrupt Channel Map Register for 672 to 672+3
0x694	CH_MAP_REG169	Interrupt Channel Map Register for 676 to 676+3
0x698	CH_MAP_REG170	Interrupt Channel Map Register for 680 to 680+3
0x69C	CH_MAP_REG171	Interrupt Channel Map Register for 684 to 684+3
0x800	HINT_MAP_REG0	Host Interrupt Map Register for 0 to 0+3
0x804	HINT_MAP_REG1	Host Interrupt Map Register for 4 to 4+3
0x808	HINT_MAP_REG2	Host Interrupt Map Register for 8 to 8+3
0x80C	HINT_MAP_REG3	Host Interrupt Map Register for 12 to 12+3
0x810	HINT_MAP_REG4	Host Interrupt Map Register for 16 to 16+3
0x814	HINT_MAP_REG5	Host Interrupt Map Register for 20 to 20+3
0x818	HINT_MAP_REG6	Host Interrupt Map Register for 24 to 24+3
0x81C	HINT_MAP_REG7	Host Interrupt Map Register for 28 to 28+3
0x820	HINT_MAP_REG8	Host Interrupt Map Register for 32 to 32+3
0x824	HINT_MAP_REG9	Host Interrupt Map Register for 36 to 36+3
0x828	HINT_MAP_REG10	Host Interrupt Map Register for 40 to 40+3
0x82C	HINT_MAP_REG11	Host Interrupt Map Register for 44 to 44+3
0x830	HINT_MAP_REG12	Host Interrupt Map Register for 48 to 48+3
0x834	HINT_MAP_REG13	Host Interrupt Map Register for 52 to 52+3
0x838	HINT_MAP_REG14	Host Interrupt Map Register for 56 to 56+3
0x83C	HINT_MAP_REG15	Host Interrupt Map Register for 60 to 60+3
0x840	HINT_MAP_REG16	Host Interrupt Map Register for 63 to 63+3
0x844	HINT_MAP_REG17	Host Interrupt Map Register for 66 to 66+3
0x848	HINT_MAP_REG18	Host Interrupt Map Register for 68 to 68+3
0x84C	HINT_MAP_REG19	Host Interrupt Map Register for 72 to 72+3
0x850	HINT_MAP_REG20	Host Interrupt Map Register for 76 to 76+3
0x854	HINT_MAP_REG21	Host Interrupt Map Register for 80 to 80+3
0x858	HINT_MAP_REG22	Host Interrupt Map Register for 84 to 84+3
0x85C	HINT_MAP_REG23	Host Interrupt Map Register for 88 to 88+3
0x860	HINT_MAP_REG24	Host Interrupt Map Register for 92 to 92+3
0x864	HINT_MAP_REG25	Host Interrupt Map Register for 94 to 94+3
0x868	HINT_MAP_REG26	Host Interrupt Map Register for 96 to 96+3
0x86C	HINT_MAP_REG27	Host Interrupt Map Register for 100 to 100+3
0x1500	ENABLE_HINT_REG0	Host Int Enable Register 0
0x1504	ENABLE_HINT_REG1	Host Int Enable Register 1
0x1508	ENABLE_HINT_REG2	Host Int Enable Register 2
0x150C	ENABLE_HINT_REG3	Host Int Enable Register 3

6.4 Enhanced Direct Memory Access (EDMA3) Controller

The primary purpose of the EDMA3 is to service user-programmed data transfers between two memory-mapped slave endpoints on the device. The EDMA3 services software-driven paging transfers (e.g., data movement between external memory and internal memory), performs sorting or subframe extraction of various data structures, services event driven peripherals, and offloads data transfers from the device ARM CorePac.

There are 5 EDMA channel controllers on the device:

- EDMA3CC0 has two transfer controllers: TPTC0 and TPTC1

- EDMA3CC1 has four transfer controllers: TPTC0, TPTC1, TPTC2, and TPTC3
- EDMA3CC2 has four transfer controllers: TPTC0, TPTC1, TPTC2, and TPTC3
- EDMA3CC3 has two transfer controllers: TPTC0 and TPTC1
- EDMA3CC4 has two transfer controllers: TPTC0 and TPTC1

In the context of this document, TPTCx is associated with EDMA3CCy, and is referred to as EDMA3CCy TPTCx. Each of the transfer controllers has a direct connection to the switch fabric. Section 7.2 lists the peripherals that can be accessed by the transfer controllers.

EDMA3CC0 is optimized to be used for transfers to/from/within the MSMC and DDR3 subsytems. The others are used for the remaining traffic.

Each EDMA3 channel controller includes the following features:

- Fully orthogonal transfer description
  - 3 transfer dimensions:
    - Array (multiple bytes)
    - Frame (multiple arrays)
    - Block (multiple frames)
  - Single event can trigger transfer of array, frame, or entire block
  - Independent indexes on source and destination
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes
  - Linking mechanism allows for ping-pong buffering, circular buffering, and repetitive/continuous transfers, all with no CPU intervention
  - Chaining allows multiple transfers to execute with one event
- 512 PaRAM entries for all EDMA3CC
  - Used to define transfer context for channels
  - Each PaRAM entry can be used as a DMA entry, QDMA entry, or link entry
- 64 DMA channels for all EDMA3CC
  - Manually triggered (CPU writes to channel controller register)
  - External event triggered
  - Chain triggered (completion of one transfer triggers another)
- 8 Quick DMA (QDMA) channels per EDMA3CCx
  - Used for software-driven transfers
  - Triggered upon writing to a single PaRAM set entry
- Two transfer controllers and two event queues with programmable system-level priority for EDMA3CC0, EDMA3CC3 and EDMA3CC4
- Four transfer controllers and four event queues with programmable system-level priority for each of EDMA3CC1 and EDMA3CC2
- Interrupt generation for transfer completion and error conditions
- Debug visibility
  - Queue watermarking/threshold allows detection of maximum usage of event queues
  - Error and status recording to facilitate debug

6.4.1 EDMA3 Device-Specific Information

The EDMA supports two addressing modes: constant addressing and increment addressing mode. Constant addressing mode is applicable to a very limited set of use cases. For most applications increment mode can be used. For more information on these two addressing modes, see the *KeyStone Architecture Enhanced Direct Memory Access 3 (EDMA3) User's Guide* (SPRUGS5).

For the range of memory addresses that includes EDMA3 channel controller (EDMA3CC) control registers and EDMA3 transfer controller (TPTC) control registers, see Section Section 6.1. For memory offsets and other details on EDMA3CC and TPTC Control Register entries, see the *KeyStone Architecture Enhanced Direct Memory Access 3 (EDMA3) User's Guide* (SPRUGS5).

6.4.2 EDMA3 Channel Controller Configuration

Table 6-25 shows the configuration for each of the EDMA3 channel controllers present on the device.

Table 6-25. EDMA3 Channel Controller Configuration

DESCRIPTION	EDMA3 CC0	EDMA3 CC1	EDMA3 CC2	EDMA3 CC3	EDMA3 CC4
Number of DMA channels in channel controller	64	64	64	64	64
Number of QDMA channels	8	8	8	8	8
Number of interrupt channels	64	64	64	64	64
Number of PaRAM set entries	512	512	512	512	512
Number of event queues	2	4	4	2	2
Number of transfer controllers	2	4	4	2	2
Memory protection existence	Yes	Yes	Yes	Yes	Yes
Number of memory protection and shadow regions	8	8	8	8	8

6.4.3 EDMA3 Transfer Controller Configuration

Each transfer controller on the device is designed differently based on considerations like performance requirements, system topology (like main TeraNet bus width, external memory bus width), etc. The parameters that determine the transfer controller configurations are:

- **FIFOSIZE:** Determines the size in bytes for the data FIFO that is the temporary buffer for the in-flight data. The data FIFO is where the read return data read by the TC read controller from the source endpoint is stored and subsequently written out to the destination endpoint by the TC write controller.
- **BUSWIDTH:** The width of the read and write data buses in bytes, for the TC read and write controller, respectively. This is typically equal to the bus width of the main TeraNet interface.
- **Default Burst Size (DBS):** The DBS is the maximum number of bytes per read/write command issued by a transfer controller.
- **DSTREGDEPTH:** This determines the number of destination FIFO register sets. The number of destination FIFO register sets for a transfer controller determines the maximum number of outstanding transfer requests.

All four parameters listed above are fixed by the design of the device.

Table 6-26 shows the configuration of each of the EDMA3 transfer controllers present on the device.

Table 6-26. EDMA3 Transfer Controller Configuration

	EDMA3 CC0/CC4		EDMA3 CC1				EDMA3 CC2				EDMA3CC3	
PARAMETER	TC0	TC1	TC0	TC1	TC2	TC3	TC0	TC1	TC2	TC3	TC0	TC1
FIFOSIZE	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes	1024 bytes
BUSWIDTH	32 bytes	32 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes	16 bytes
DSTREGDEPTH	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries	4 entries
DBS	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	128 bytes	64 bytes	64 bytes

6.4.4 EDMA3 Channel Synchronization Events

The EDMA3 supports up to 64 DMA channels for all EDMA3CC that can be used to service system peripherals and to move data between system memories. DMA channels can be triggered by synchronization events generated by system peripherals. The following tables list the source of the synchronization event associated with each of the EDMA3CC DMA channels. On the AM5K2E0x, the association of each synchronization event and DMA channel is fixed and cannot be reprogrammed.

For more detailed information on the EDMA3 module and how EDMA3 events are enabled, captured, processed, prioritized, linked, chained, and cleared, etc., see the *KeyStone Architecture Enhanced Direct Memory Access 3 (EDMA3) User's Guide* (SPRUGS5).

Table 6-27. EDMA3CC0 Events for AM5K2E0x

EVENT NO.	EVENT NAME	DESCRIPTION
0	TIMER_8_INTL	Timer interrupt low
1	TIMER_8_INTH	Timer interrupt high
2	TIMER_9_INTL	Timer interrupt low
3	TIMER_9_INTH	Timer interrupt high
4	TIMER_10_INTL	Timer interrupt low
5	TIMER_10_INTH	Timer interrupt high
6	TIMER_11_INTL	Timer interrupt low
7	TIMER_11_INTH	Timer interrupt high
8	CIC_2_OUT66	CIC2 Interrupt Controller output
9	CIC_2_OUT67	CIC2 Interrupt Controller output
10	CIC_2_OUT68	CIC2 Interrupt Controller output
11	CIC_2_OUT69	CIC2 Interrupt Controller output
12	CIC_2_OUT70	CIC2 Interrupt Controller output
13	CIC_2_OUT71	CIC2 Interrupt Controller output
14	CIC_2_OUT72	CIC2 Interrupt Controller output
15	CIC_2_OUT73	CIC2 Interrupt Controller output
16	GPIO_INT8	GPIO interrupt
17	GPIO_INT9	GPIO interrupt
18	GPIO_INT10	GPIO interrupt
19	GPIO_INT11	GPIO interrupt
20	GPIO_INT12	GPIO interrupt
21	GPIO_INT13	GPIO interrupt
22	GPIO_INT14	GPIO interrupt
23	GPIO_INT15	GPIO interrupt
24	TIMER_16_INTL	Timer interrupt low
25	TIMER_16_INTH	Timer interrupt high
26	TIMER_17_INTL	Timer interrupt low
27	TIMER_17_INTH	Timer interrupt high
28	TIMER_18_INTL	Timer interrupt low
29	TIMER_18_INTH	Timer interrupt high
30	TIMER_19_INTL	Timer interrupt low
31	TIMER_19_INTH	Timer interrupt high
32	GPIO_INT0	GPIO interrupt
33	GPIO_INT1	GPIO interrupt
34	GPIO_INT2	GPIO interrupt
35	GPIO_INT3	GPIO interrupt
36	GPIO_INT4	GPIO interrupt

Table 6-27. EDMA3CC0 Events for AM5K2E0x (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
37	GPIO_INT5	GPIO interrupt
38	GPIO_INT6	GPIO interrupt
39	GPIO_INT7	GPIO interrupt
40	Reserved	Reserved
41	Reserved	Reserved
42	TIMER_12_INTL	Timer interrupt low
43	TIMER_12_INTH	Timer interrupt high
44	TIMER_13_INTL	Timer interrupt low
45	TIMER_13_INTH	Timer interrupt high
46	Reserved	Reserved
47	SEM_INT8	Semaphore interrupt
48	SEM_INT9	Semaphore interrupt
49	SEM_INT10	Semaphore interrupt
50	SEM_INT11	Semaphore interrupt
51	SEM_INT12	Semaphore interrupt
52	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
53	ARM_TBR_DMA	ARM trace buffer (TBR) DMA event
54	QMSS_QUE_PEND_560	Navigator transmit queue pending event for indicated queue
55	QMSS_QUE_PEND_561	Navigator transmit queue pending event for indicated queue
56	QMSS_QUE_PEND_562	Navigator transmit queue pending event for indicated queue
57	QMSS_QUE_PEND_563	Navigator transmit queue pending event for indicated queue
58	QMSS_QUE_PEND_564	Navigator transmit queue pending event for indicated queue
59	QMSS_QUE_PEND_565	Navigator transmit queue pending event for indicated queue
60	QMSS_QUE_PEND_566	Navigator transmit queue pending event for indicated queue
61	QMSS_QUE_PEND_567	Navigator transmit queue pending event for indicated queue
62	QMSS_QUE_PEND_568	Navigator transmit queue pending event for indicated queue
63	QMSS_QUE_PEND_569	Navigator transmit queue pending event for indicated queue

Table 6-28. EDMA3CC1 Events for AM5K2E0x

EVENT NO.	EVENT NAME	DESCRIPTION
0	GPIO_INT28	GPIO interrupt
1	GPIO_INT29	GPIO interrupt
2	SPI_0_XEVT	SPI0 transmit event
3	SPI_0_REVT	SPI0 receive event
4	SEM_INT8	Semaphore interrupt
5	SEM_INT9	Semaphore interrupt
6	GPIO_INT0	GPIO interrupt
7	GPIO_INT1	GPIO interrupt
8	GPIO_INT2	GPIO interrupt
9	GPIO_INT3	GPIO interrupt
10	QMSS_QUE_PEND_570	Navigator transmit queue pending event for indicated queue
11	QMSS_QUE_PEND_571	Navigator transmit queue pending event for indicated queue
12	QMSS_QUE_PEND_572	Navigator transmit queue pending event for indicated queue
13	QMSS_QUE_PEND_573	Navigator transmit queue pending event for indicated queue
14	Reserved	Reserved
15	QMSS_QUE_PEND_574	Navigator transmit queue pending event for indicated queue
16	QMSS_QUE_PEND_575	Navigator transmit queue pending event for indicated queue

Table 6-28. EDMA3CC1 Events for AM5K2E0x (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
17	QMSS_QUE_PEND_576	Navigator transmit queue pending event for indicated queue
18	QMSS_QUE_PEND_577	Navigator transmit queue pending event for indicated queue
19	QMSS_QUE_PEND_578	Navigator transmit queue pending event for indicated queue
20	QMSS_QUE_PEND_579	Navigator transmit queue pending event for indicated queue
21	QMSS_QUE_PEND_580	Navigator transmit queue pending event for indicated queue
22	TIMER_8_INTL	Timer interrupt low
23	TIMER_8_INTH	Timer interrupt high
24	TIMER_9_INTL	Timer interrupt low
25	TIMER_9_INTH	Timer interrupt high
26	TIMER_10_INTL	Timer interrupt low
27	TIMER_10_INTH	Timer interrupt high
28	TIMER_11_INTL	Timer interrupt low
29	TIMER_11_INTH	Timer interrupt high
30	TIMER_12_INTL	Timer interrupt low
31	TIMER_12_INTH	Timer interrupt high
32	TIMER_13_INTL	Timer interrupt low
33	TIMER_13_INTH	Timer interrupt high
34	TIMER_14_INTL	Timer interrupt low
35	TIMER_14_INTH	Timer interrupt high
36	TIMER_15_INTL	Timer interrupt low
37	TIMER_15_INTH	Timer interrupt high
38	SEM_INT10	Semaphore interrupt
39	SEM_INT11	Semaphore interrupt
40	SEM_INT12	Semaphore interrupt
41	SR_0_SR_TEMPSENSOR	SmartReflex temperature threshold crossing interrupt
42	TSIP_RCV_FINT0	TSIP receive frame interrupt for Channel 0
43	TSIP_XMT_FINT0	TSIP transmit frame interrupt for Channel 0
44	TSIP_RCV_SFINT0	TSIP receive super frame interrupt for Channel 0
45	TSIP_XMT_SFINT0	TSIP transmit super frame interrupt for Channel 0
46	TSIP_RCV_FINT1	TSIP receive frame interrupt for Channel 1
47	TSIP_XMT_FINT1	TSIP transmit frame interrupt for Channel 1
48	TSIP_RCV_SFINT1	TSIP receive super frame interrupt for Channel 1
49	TSIP_XMT_SFINT1	TSIP transmit super frame interrupt for Channel 1
50	CIC_2_OUT8	CIC2 Interrupt Controller output
51	GPIO_INT30	GPIO interrupt
52	GPIO_INT31	GPIO interrupt
53	I2C_0_REVT	I2C0 receive
54	I2C_0_XEVT	I2C0 transmit
55	CIC_2_OUT13	CIC2 Interrupt Controller output
56	CIC_2_OUT14	CIC2 Interrupt Controller output
57	CIC_2_OUT15	CIC2 Interrupt Controller output
58	CIC_2_OUT16	CIC2 Interrupt Controller output
59	CIC_2_OUT17	CIC2 Interrupt Controller output
60	CIC_2_OUT18	CIC2 Interrupt Controller output
61	CIC_2_OUT19	CIC2 Interrupt Controller output
62	Reserved	Reserved
63	Reserved	Reserved

Table 6-29. EDMA3CC2 Events for AM5K2E0x

EVENT NO.	EVENT NAME	DESCRIPTION
0	UART_1_URXEVT	UART1 receive event
1	UART_1_UTXEVT	UART1 transmit event
2	SPI_1_XEVT	SPI1 receive event
3	SPI_1_REVT	SPI1 transmit event
4	SPI_2_XEVT	SPI2 receive event
5	SPI_2_REVT	SPI2 transmit event
6	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
7	ARM_TBR_DMA	ARM trace buffer (TBR) DMA event
8	Reserved	Reserved
9	Reserved	Reserved
10	I2C_1_REVT	I2C1 receive
11	I2C_1_XEVT	I2C1 transmit
12	I2C_2_REVT	I2C2 receive
13	I2C_2_XEVT	I2C2 transmit
14	GPIO_INT16	GPIO interrupt
15	GPIO_INT17	GPIO interrupt
16	GPIO_INT18	GPIO interrupt
17	GPIO_INT19	GPIO interrupt
18	GPIO_INT20	GPIO interrupt
19	GPIO_INT21	GPIO interrupt
20	GPIO_INT22	GPIO interrupt
21	GPIO_INT23	GPIO interrupt
22	GPIO_INT24	GPIO interrupt
23	GPIO_INT25	GPIO interrupt
24	GPIO_INT26	GPIO interrupt
25	GPIO_INT27	GPIO interrupt
26	GPIO_INT0	GPIO interrupt
27	GPIO_INT1	GPIO interrupt
28	GPIO_INT2	GPIO interrupt
29	GPIO_INT3	GPIO interrupt
30	GPIO_INT4	GPIO interrupt
31	GPIO_INT5	GPIO interrupt
32	GPIO_INT6	GPIO interrupt
33	GPIO_INT7	GPIO interrupt
34	ARM_NCNTVIRQ3	ARM virtual timer interrupt for core 3
35	ARM_NCNTVIRQ2	ARM virtual timer interrupt for core 2
36	ARM_NCNTVIRQ1	ARM virtual timer interrupt for core 1
37	ARM_NCNTVIRQ0	ARM virtual timer interrupt for core 0
38	CIC_2_OUT48	CIC2 Interrupt Controller output
39	Reserved	Reserved
40	UART_0_URXEVT	UART0 receive event
41	UART_0_UTXEVT	UART0 transmit event
42	CIC_2_OUT22	CIC2 Interrupt Controller output
43	CIC_2_OUT23	CIC2 Interrupt Controller output
44	CIC_2_OUT24	CIC2 Interrupt Controller output
45	CIC_2_OUT25	CIC2 Interrupt Controller output
46	CIC_2_OUT26	CIC2 Interrupt Controller output

Table 6-29. EDMA3CC2 Events for AM5K2E0x (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
47	CIC_2_OUT27	CIC2 Interrupt Controller output
48	CIC_2_OUT28	CIC2 Interrupt Controller output
49	SPI_0_XEVT	SPI0 receive event
50	SPI_0_REVT	SPI0 transmit event
51	Reserved	Reserved
52	ARM_NCNTPNSIRQ3	ARM non secure timer interrupt for Core 3
53	ARM_NCNTPNSIRQ2	ARM non secure timer interrupt for Core 2
54	ARM_NCNTPNSIRQ1	ARM non secure timer interrupt for Core 1
55	ARM_NCNTPNSIRQ0	ARM non secure timer interrupt for Core 0
56	QMSS_QUE_PEND_581	Navigator transmit queue pending event for indicated queue
57	QMSS_QUE_PEND_582	Navigator transmit queue pending event for indicated queue
58	QMSS_QUE_PEND_583	Navigator transmit queue pending event for indicated queue
59	QMSS_QUE_PEND_584	Navigator transmit queue pending event for indicated queue
60	QMSS_QUE_PEND_585	Navigator transmit queue pending event for indicated queue
61	QMSS_QUE_PEND_586	Navigator transmit queue pending event for indicated queue
62	QMSS_QUE_PEND_587	Navigator transmit queue pending event for indicated queue
63	QMSS_QUE_PEND_588	Navigator transmit queue pending event for indicated queue

Table 6-30. EDMA3CC3 Events for AM5K2E0x

EVENT NO.	EVENT NAME	DESCRIPTION
0	Reserved	Reserved
1	Reserved	Reserved
2	SPI_2_XEVT	SPI2 transmit event
3	SPI_2_REVT	SPI2 receive event
4	I2C_2_REVT	I2C2 receive
5	I2C_2_XEVT	I2C2 transmit
6	UART_1_URXEVT	UART1 receive event
7	UART_1_UTXEVT	UART1 transmit event
8	Reserved	Reserved
9	Reserved	Reserved
10	SPI_1_XEVT	SPI1 transmit event
11	SPI_1_REVT	SPI1 receive event
12	I2C_0_REVT	I2C0 receive
13	I2C_0_XEVT	I2C0 transmit
14	I2C_1_REVT	I2C1 receive
15	I2C_1_XEVT	I2C1 transmit
16	TIMER_16_INTL	Timer interrupt low
17	TIMER_16_INTH	Timer interrupt high
18	TIMER_17_INTL	Timer interrupt low
19	TIMER_17_INTH	Timer interrupt high
20	ARM_TBR_DMA	Debug trace buffer (TBR) DMA event
21	DBGTBR_DMAINT	ARM trace buffer (TBR) DMA event
22	UART_0_URXEVT	UART0 receive event
23	UART_0_UTXEVT	UART0 transmit event
24	GPIO_INT16	GPIO interrupt
25	GPIO_INT17	GPIO interrupt
26	GPIO_INT18	GPIO interrupt

Table 6-30. EDMA3CC3 Events for AM5K2E0x (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
27	GPIO_INT19	GPIO interrupt
28	GPIO_INT20	GPIO interrupt
29	GPIO_INT21	GPIO interrupt
30	GPIO_INT22	GPIO interrupt
31	GPIO_INT23	GPIO interrupt
32	GPIO_INT24	GPIO interrupt
33	GPIO_INT25	GPIO interrupt
34	GPIO_INT26	GPIO interrupt
35	GPIO_INT27	GPIO interrupt
36	GPIO_INT28	GPIO interrupt
37	GPIO_INT29	GPIO interrupt
38	GPIO_INT30	GPIO interrupt
39	GPIO_INT31	GPIO interrupt
40	QMSS_QUE_PEND_589	Navigator transmit queue pending event for indicated queue
41	QMSS_QUE_PEND_590	Navigator transmit queue pending event for indicated queue
42	QMSS_QUE_PEND_591	Navigator transmit queue pending event for indicated queue
43	QMSS_QUE_PEND_592	Navigator transmit queue pending event for indicated queue
44	QMSS_QUE_PEND_593	Navigator transmit queue pending event for indicated queue
45	QMSS_QUE_PEND_594	Navigator transmit queue pending event for indicated queue
46	QMSS_QUE_PEND_595	Navigator transmit queue pending event for indicated queue
47	QMSS_QUE_PEND_596	Navigator transmit queue pending event for indicated queue
48	QMSS_QUE_PEND_597	Navigator transmit queue pending event for indicated queue
49	QMSS_QUE_PEND_598	Navigator transmit queue pending event for indicated queue
50	QMSS_QUE_PEND_599	Navigator transmit queue pending event for indicated queue
51	QMSS_QUE_PEND_600	Navigator transmit queue pending event for indicated queue
52	QMSS_QUE_PEND_601	Navigator transmit queue pending event for indicated queue
53	QMSS_QUE_PEND_602	Navigator transmit queue pending event for indicated queue
54	QMSS_QUE_PEND_603	Navigator transmit queue pending event for indicated queue
55	QMSS_QUE_PEND_604	Navigator transmit queue pending event for indicated queue
56	CIC_2_OUT57	CIC2 Interrupt Controller output
57	CIC_2_OUT50	CIC2 Interrupt Controller output
58	CIC_2_OUT51	CIC2 Interrupt Controller output
59	CIC_2_OUT52	CIC2 Interrupt Controller output
60	CIC_2_OUT53	CIC2 Interrupt Controller output
61	CIC_2_OUT54	CIC2 Interrupt Controller output
62	CIC_2_OUT55	CIC2 Interrupt Controller output
63	CIC_2_OUT56	CIC2 Interrupt Controller output

Table 6-31. EDMA3CC4 Events for AM5K2E0x

EVENT NO.	EVENT NAME	DESCRIPTION
0	GPIO_INT16	GPIO interrupt
1	GPIO_INT17	GPIO interrupt
2	GPIO_INT18	GPIO interrupt
3	GPIO_INT19	GPIO interrupt
4	GPIO_INT20	GPIO interrupt
5	GPIO_INT21	GPIO interrupt
6	GPIO_INT22	GPIO interrupt

Table 6-31. EDMA3CC4 Events for AM5K2E0x (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
7	GPIO_INT23	GPIO interrupt
8	GPIO_INT24	GPIO interrupt
9	GPIO_INT25	GPIO interrupt
10	GPIO_INT26	GPIO interrupt
11	GPIO_INT27	GPIO interrupt
12	GPIO_INT28	GPIO interrupt
13	GPIO_INT29	GPIO interrupt
14	GPIO_INT30	GPIO interrupt
15	GPIO_INT31	GPIO interrupt
16	Reserved	Reserved
17	SEM_INT8	Semaphore interrupt
18	SEM_INT9	Semaphore interrupt
19	SEM_INT10	Semaphore interrupt
20	SEM_INT11	Semaphore interrupt
21	SEM_INT12	Semaphore interrupt
22	TIMER_12_INTL	Timer interrupt low
23	TIMER_12_INTH	Timer interrupt high
24	TIMER_8_INTL	Timer interrupt low
25	TIMER_8_INTH	Timer interrupt high
26	TIMER_14_INTL	Timer interrupt low
27	TIMER_14_INTH	Timer interrupt high
28	TIMER_15_INTL	Timer interrupt low
29	TIMER_15_INTH	Timer interrupt high
30	DBGTBR_DMAINT	Debug trace buffer (TBR) DMA event
31	ARM_TBR_DMA	ARM trace buffer (TBR) DMA event
32	QMSS_QUE_PEND_658	Navigator transmit queue pending event for indicated queue
33	QMSS_QUE_PEND_659	Navigator transmit queue pending event for indicated queue
34	QMSS_QUE_PEND_660	Navigator transmit queue pending event for indicated queue
35	QMSS_QUE_PEND_661	Navigator transmit queue pending event for indicated queue
36	QMSS_QUE_PEND_662	Navigator transmit queue pending event for indicated queue
37	QMSS_QUE_PEND_663	Navigator transmit queue pending event for indicated queue
38	QMSS_QUE_PEND_664	Navigator transmit queue pending event for indicated queue
39	QMSS_QUE_PEND_665	Navigator transmit queue pending event for indicated queue
40	QMSS_QUE_PEND_605	Navigator transmit queue pending event for indicated queue
41	QMSS_QUE_PEND_606	Navigator transmit queue pending event for indicated queue
42	QMSS_QUE_PEND_607	Navigator transmit queue pending event for indicated queue
43	QMSS_QUE_PEND_608	Navigator transmit queue pending event for indicated queue
44	QMSS_QUE_PEND_609	Navigator transmit queue pending event for indicated queue
45	QMSS_QUE_PEND_610	Navigator transmit queue pending event for indicated queue
46	QMSS_QUE_PEND_611	Navigator transmit queue pending event for indicated queue
47	QMSS_QUE_PEND_612	Navigator transmit queue pending event for indicated queue
48	ARM_NCNTVIRQ3	ARM virtual timer interrupt for Core 3
49	ARM_NCNTVIRQ2	ARM virtual timer interrupt for Core 2
50	ARM_NCNTVIRQ1	ARM virtual timer interrupt for Core 1
51	ARM_NCNTVIRQ0	ARM virtual timer interrupt for Core 0
52	ARM_NCNTPNSIRQ3	ARM non secure timer interrupt for Core 3
53	ARM_NCNTPNSIRQ2	ARM non secure timer interrupt for Core 2

Table 6-31. EDMA3CC4 Events for AM5K2E0x (continued)

EVENT NO.	EVENT NAME	DESCRIPTION
54	ARM_NCNTPNSIRQ1	ARM non secure timer interrupt for Core 1
55	ARM_NCNTPNSIRQ0	ARM non secure timer interrupt for Core 0
56	CIC_2_OUT82	CIC2 Interrupt Controller output
57	CIC_2_OUT83	CIC2 Interrupt Controller output
58	CIC_2_OUT84	CIC2 Interrupt Controller output
59	CIC_2_OUT85	CIC2 Interrupt Controller output
60	CIC_2_OUT86	CIC2 Interrupt Controller output
61	CIC_2_OUT87	CIC2 Interrupt Controller output
62	CIC_2_OUT88	CIC2 Interrupt Controller output
63	CIC_2_OUT89	CIC2 Interrupt Controller output

7 System Interconnect

On the KeyStone II devices, the EDMA3 transfer controllers and the system peripherals are interconnected through the TeraNets, which are non-blocking switch fabrics enabling fast and contention-free internal data movement. The TeraNets provide low-latency, concurrent data transfers between master peripherals and slave peripherals. The TeraNets also allow for seamless arbitration between the system masters when accessing system slaves.

The ARM CorePac is connected to the MSMC and the debug subsystem directly, and to other masters via the TeraNets. Through the MSMC, the ARM CorePacs can be interconnected to DDR3 and TeraNet 3_A, which allows the ARM CorePacs to access to the peripheral buses:

- TeraNet 3P_A for peripheral configuration
- TeraNet 6P_A for ARM Boot ROM

7.1 Internal Buses and Switch Fabrics

The the ARM CorePacs, the EDMA3 traffic controllers, and the various system peripherals can be classified into two categories: masters and slaves.

- **Masters** are capable of initiating read and write transfers in the system and do not rely on the EDMA3 for their data transfers.
- **Slaves** on the other hand rely on the masters to perform transfers to and from them.

Examples of masters include the EDMA3 traffic controllers and network coprocessor packet DMA.

Examples of slaves include the SPI, UART, and I^2C.

The masters and slaves in the device communicate through the TeraNet (switch fabric). The device contains two types of switch fabric:

- **Data** TeraNet is a high-throughput interconnect mainly used to move data across the system
- **Configuration** TeraNet is mainly used to access peripheral registers

Some peripherals have both a data bus and a configuration bus interface, while others only have one type of interface. Furthermore, the bus interface width and speed varies from peripheral to peripheral.

Note that the data TeraNet also connects to the configuration TeraNet.

7.2 Switch Fabric Connections Matrix - Data Space

The figures below show the connections between masters and slaves through various sections of the TeraNet.

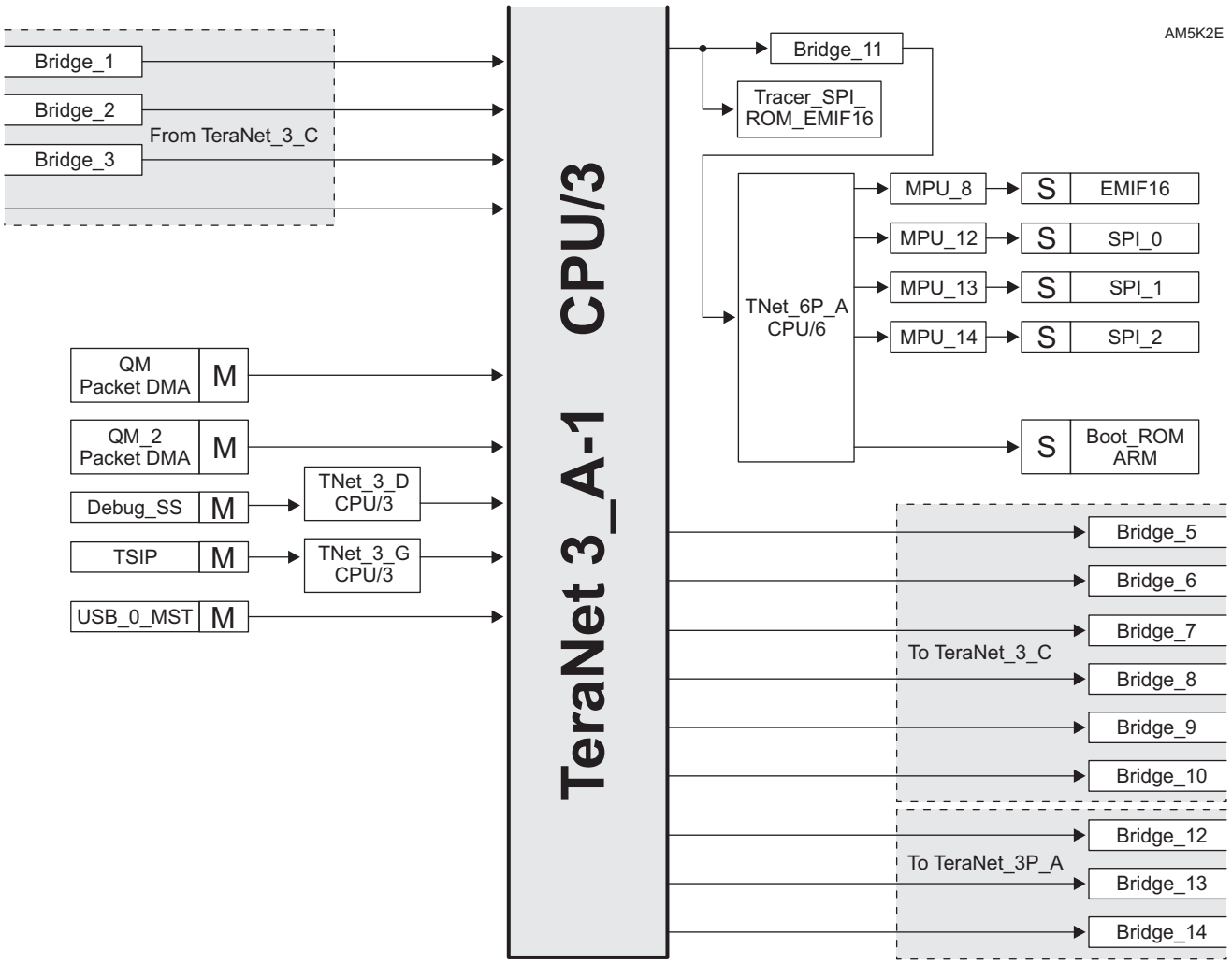


Figure 7-1. TeraNet 3_A-1

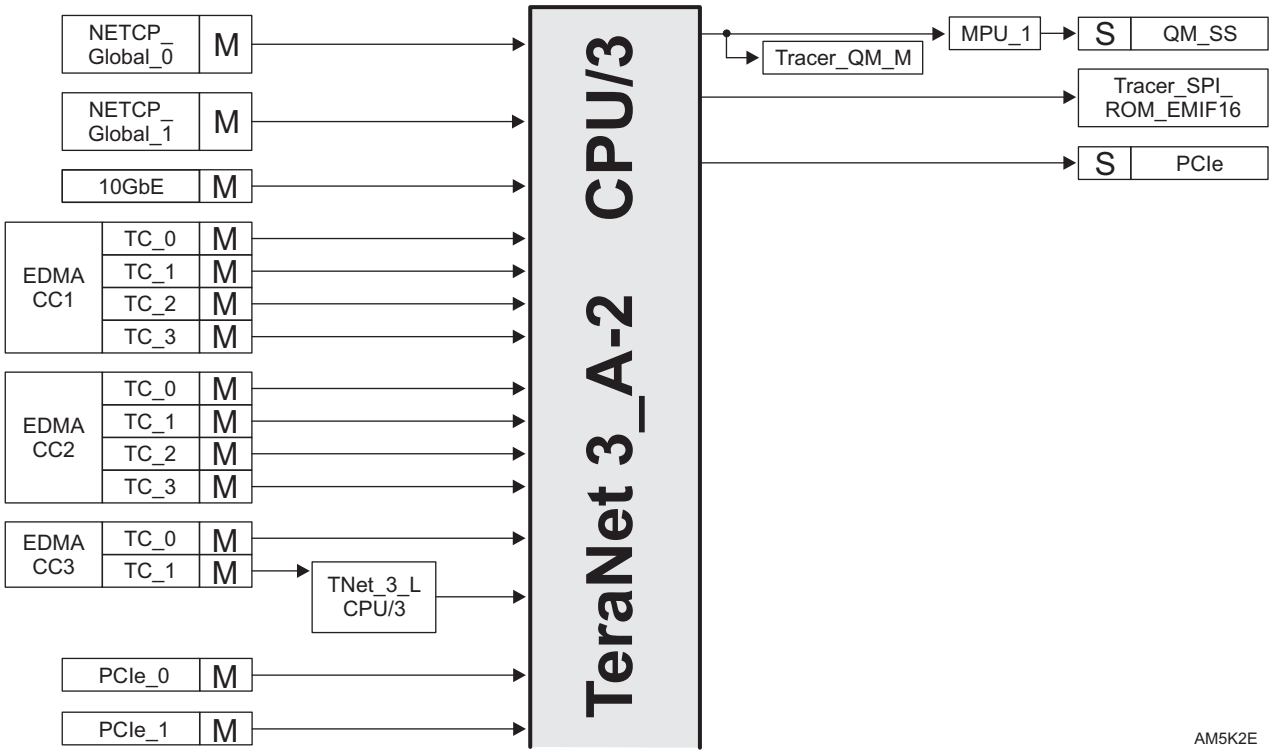


Figure 7-2. TeraNet 3_A-2

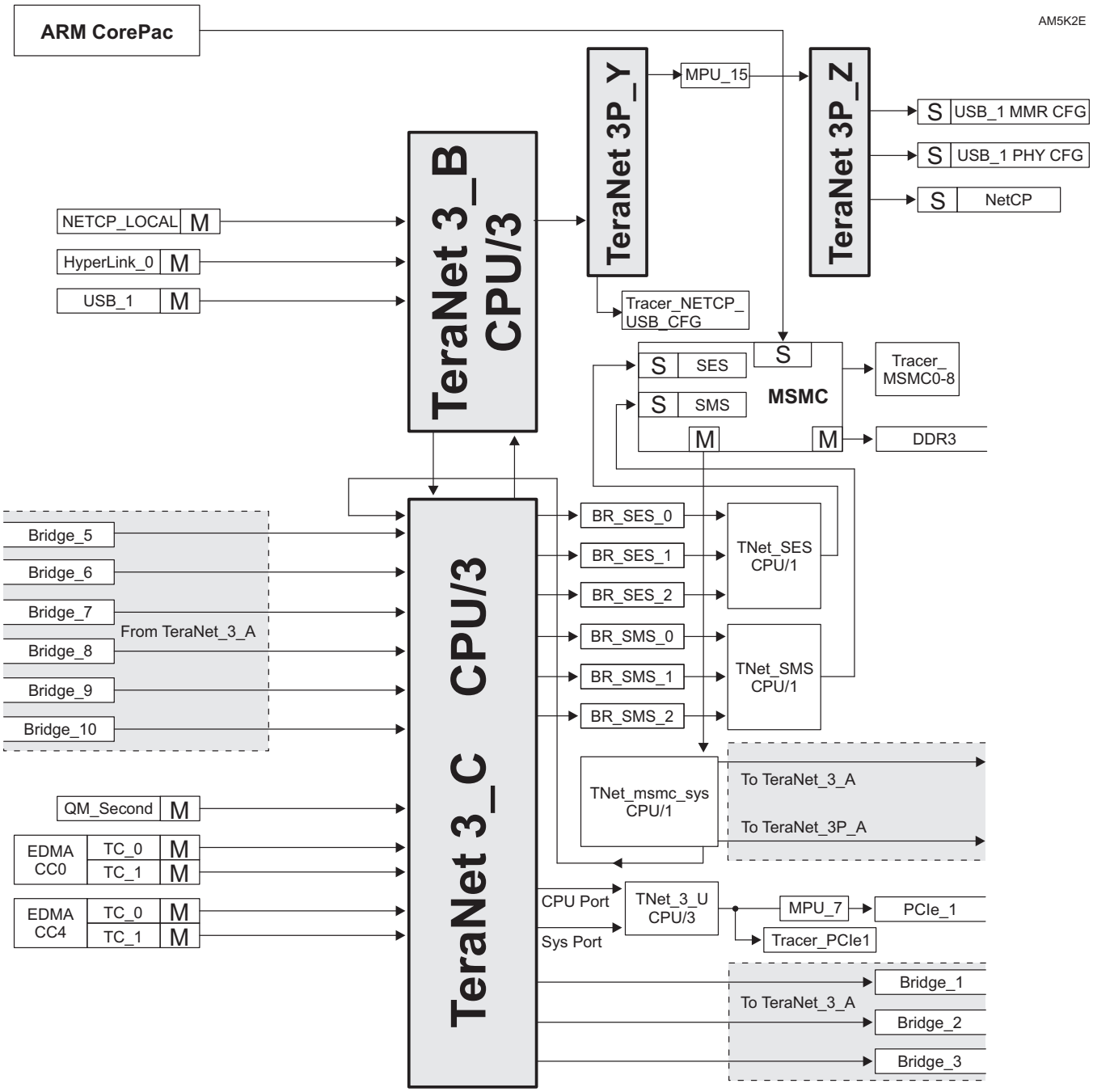


Figure 7-3. TeraNet 3_C

The following table lists the master and slave end-point connections.

Intersecting cells may contain one of the following:

- **Y** — There is a connection between this master and that slave.
- **-** — There is NO connection between this master and that slave.
- ***n*** — A numeric value indicates that the path between this master and that slave goes through bridge *n*.

Table 7-1. AM5K2E04/02 Data Space Interconnect

	Slaves									
MASTERS	**AEMIF16**	**BootROM_ARM**	**DBG_STM**	**HyperLink0**	**MSMC_SES**	**MSMC_SMS**	**PCIE0**	**PCIE1**	**QM**	**SPI(0-2)**
10GbE	-	-	-	-	SES_2	SMS_2	Y	Y	Y	-
CPT_CFG	-	-	Y	-	-	-	-	-	-	-
CPT_DDR3	-	-	Y	-	-	-	-	-	-	-
CPT_INTC	-	-	Y	-	-	-	-	-	-	-
CPT_MSMC(0-7)	-	-	Y	-	-	-	-	-	-	-
CPT_QM_CFG1	-	-	Y	-	-	-	-	-	-	-
CPT_QM_CFG2	-	-	Y	-	-	-	-	-	-	-
CPT_QM_M	-	-	Y	-	-	-	-	-	-	-
CPT_SPI_ROM_EMIF16	-	-	Y	-	-	-	-	-	-	-
CPT_TPCC(0_4)T	-	-	Y	-	-	-	-	-	-	-
CPT_TPCC(1_2_3)T	-	-	Y	-	-	-	-	-	-	-
DBG_DAP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
TSIP_DMA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA0_CC_TR	-	-	-	-	-	-	-	-	-	-
EDMA0_TC0_RD	2, 11	2, 11	-	Y	SES_0	SMS_0	Y	Y	Y	2, 11
EDMA0_TC0_WR	2, 11	-	-	Y	SES_0	SMS_0	Y	Y	Y	2,11
EDMA0_TC1_RD	3, 11	3, 11	-	Y	SES_1	SMS_1	Y	Y	-	3, 11
EDMA0_TC1_WR	3, 11	-	-	Y	SES_1	SMS_1	Y	Y	-	3, 11
EDMA1_CC_TR	-	-	-	-	-	-	-	-	-	-
EDMA1_TC0_RD	11	11	-	Y	SES_0	SMS_0	Y	Y	Y	11
EDMA1_TC0_WR	11	-	Y	Y	SES_0	SMS_0	Y	Y	Y	11
EDMA1_TC1_RD	11	Y	-	Y	SES_1	SMS_1	Y	Y	Y	11
EDMA1_TC1_WR	11	-	-	Y	SES_1	SMS_1	Y	Y	Y	11
EDMA1_TC2_RD	11	Y	-	Y	SES_1	SMS_1	Y	Y	-	11
EDMA1_TC2_WR	11	-	-	Y	SES_1	SMS_1	Y	Y	-	11
EDMA1_TC3_RD	11	Y	-	Y	SES_1	SMS_1	Y	Y	-	11
EDMA1_TC3_WR	11	-	Y	Y	SES_1	SMS_1	Y	Y	-	11
EDMA2_CC_TR	-	-	-	-	-	-	-	-	-	-
EDMA2_TC0_RD	11	Y	-	Y	SES_2	SMS_2	Y	Y	Y	11
EDMA2_TC0_WR	11	-	Y	Y	SES_2	SMS_2	Y	Y	Y	11
EDMA2_TC1_RD	11	Y	-	Y	SES_2	SMS_2	Y	Y	Y	11
EDMA2_TC1_WR	11	-	-	Y	SES_2	SMS_2	Y	Y	Y	11
EDMA2_TC2_RD	11	Y	-	Y	SES_0	SMS_0	Y	Y	-	11
EDMA2_TC2_WR	11	-	Y	Y	SES_0	SMS_0	Y	Y	-	11
EDMA2_TC3_RD	11	Y	-	Y	SES_0	SMS_0	Y	Y	-	11
EDMA2_TC3_WR	11	-	-	Y	SES_0	SMS_0	Y	Y	-	11

Table 7-1. AM5K2E04/02 Data Space Interconnect (continued)

MASTERS	Slaves									
	AEMIF16	BootROM_ARM	DBG_STM	HyperLink0	MSMC_SES	MSMC_SMS	PCIE0	PCIE1	QM	SPI(0-2)
EDMA3_CC_TR	-	-	-	-	-	-	-	-	-	-
EDMA3_TC0_RD	11	Y	-	Y	SES_1	SMS_1	Y	Y	Y	11
EDMA3_TC0_WR	11	-	Y	Y	SES_1	SMS_1	Y	Y	Y	11
EDMA3_TC1_RD	11	Y	-	Y	SES_1	SMS_1	Y	Y	-	11
EDMA3_TC1_WR	11	-	-	Y	SES_1	SMS_1	Y	Y	-	11
EDMA4_CC_TR	-	-	-	-	-	-	-	-	-	-
EDMA4_TC0_RD	2, 11	2, 11	-	Y	SES_1	SMS_1	Y	Y	Y	2, 11
EDMA4_TC0_WR	2, 11	-	-	Y	SES_1	SMS_1	Y	Y	Y	2, 11
EDMA4_TC1_RD	3, 11	3, 11	-	Y	SES_1	SMS_1	Y	Y	-	3, 11
EDMA4_TC1_WR	3, 11	-	-	Y	SES_1	SMS_1	Y	Y	-	3, 11
HyperLink0_Master	11	1, 11	-	-	Y	Y	Y	Y	Y	Y
MSMC_SYS	11	11	Y	Y	-	-	Y	Y	Y	11
NETCP	-	-	-	-	SES_1	SMS_1	Y	Y	Y	-
PCIE0	11	-	Y	10	SES_2	SMS_2	-	-	Y	11
PCIE1	11	-	Y	10	SES_2	SMS_2	-	-	Y	11
QM_Master1	-	-	-	Y	SES_0	SMS_0	-	-	Y	-
QM_Master2	-	-	-	Y	SES_1	SMS_1	-	-	Y	-
QM_SEC	-	-	Y	Y	SES_2	SMS_2	-	-	-	-
USB0	-	-	Y	Y	SES_0	SMS_0	-	-	Y	-
USB1	-	-	Y	Y	SES_0	SMS_0	-	-	Y	-

7.3 Switch Fabric Connections Matrix - Configuration Space

The figures below show the connections between masters and slaves through various sections of the TeraNet.

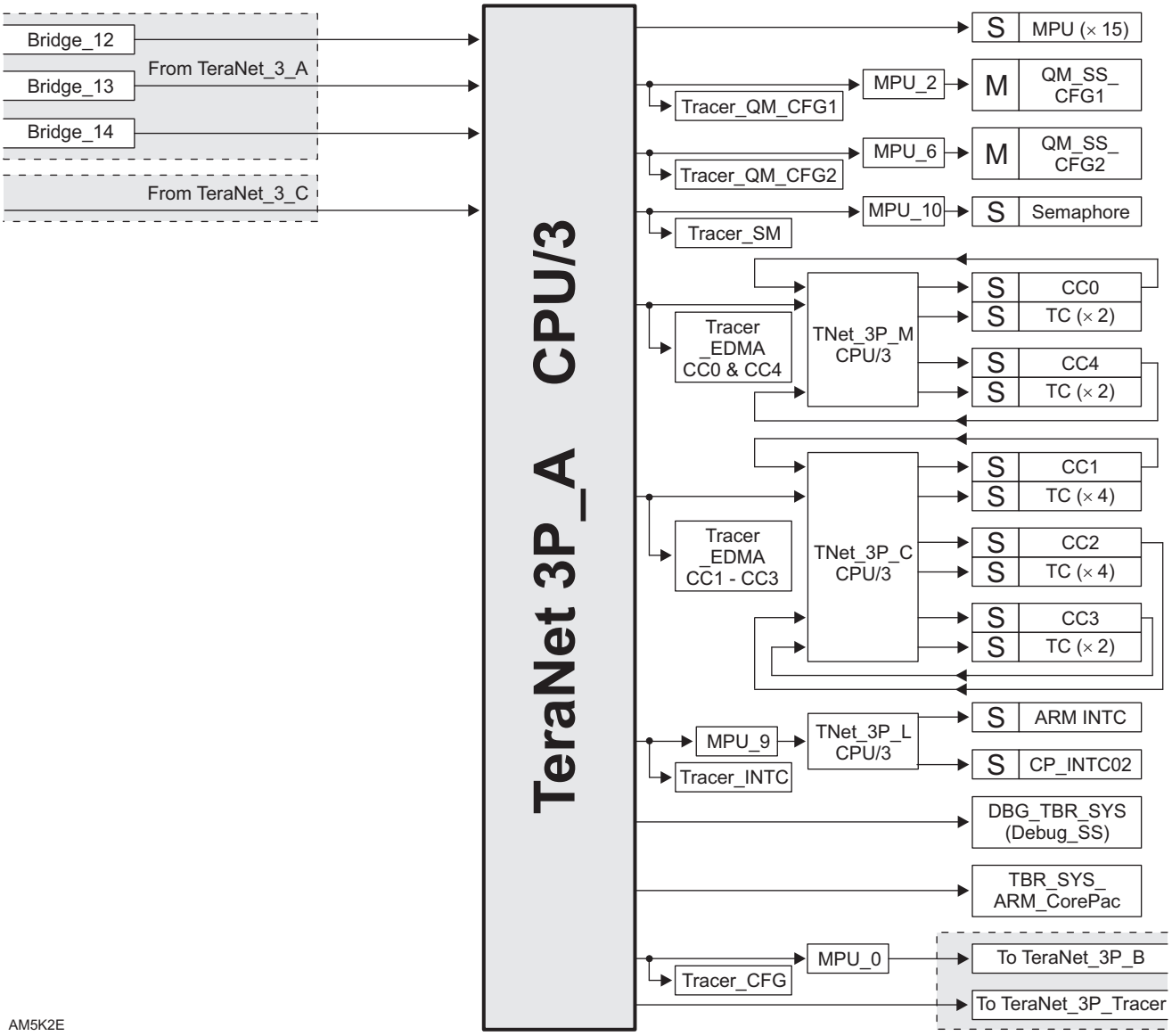


Figure 7-4. TeraNet 3P_A

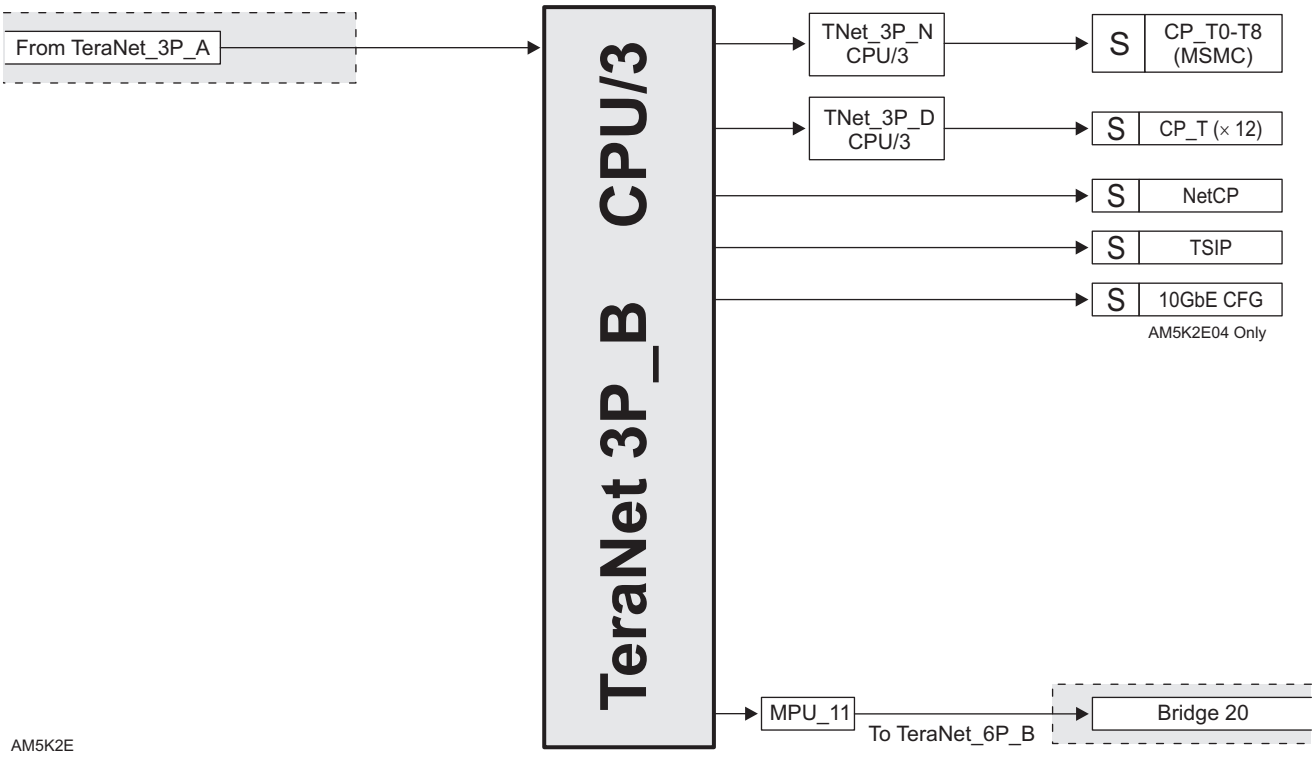


Figure 7-5. TeraNet 3P_B

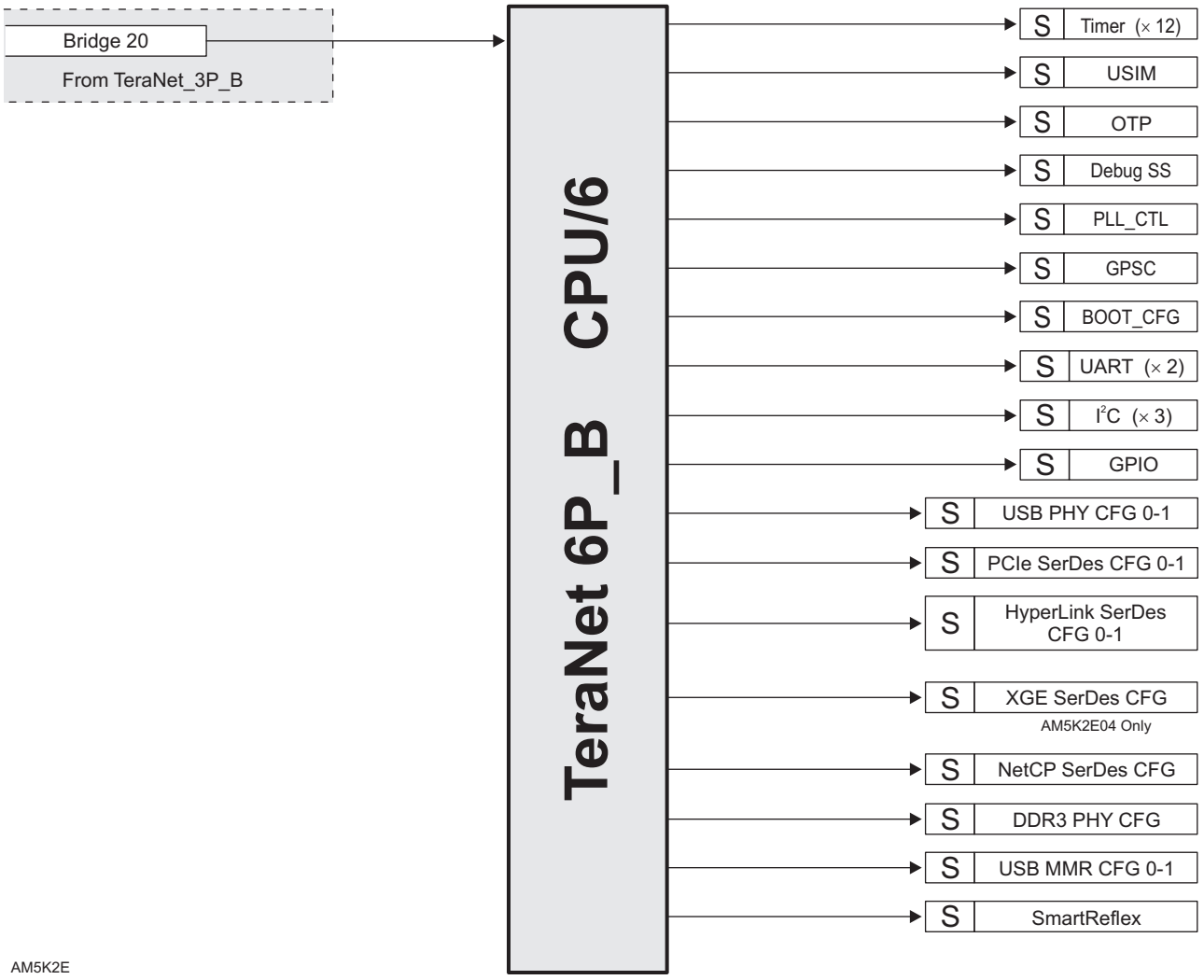


Figure 7-6. TeraNet 6P_B

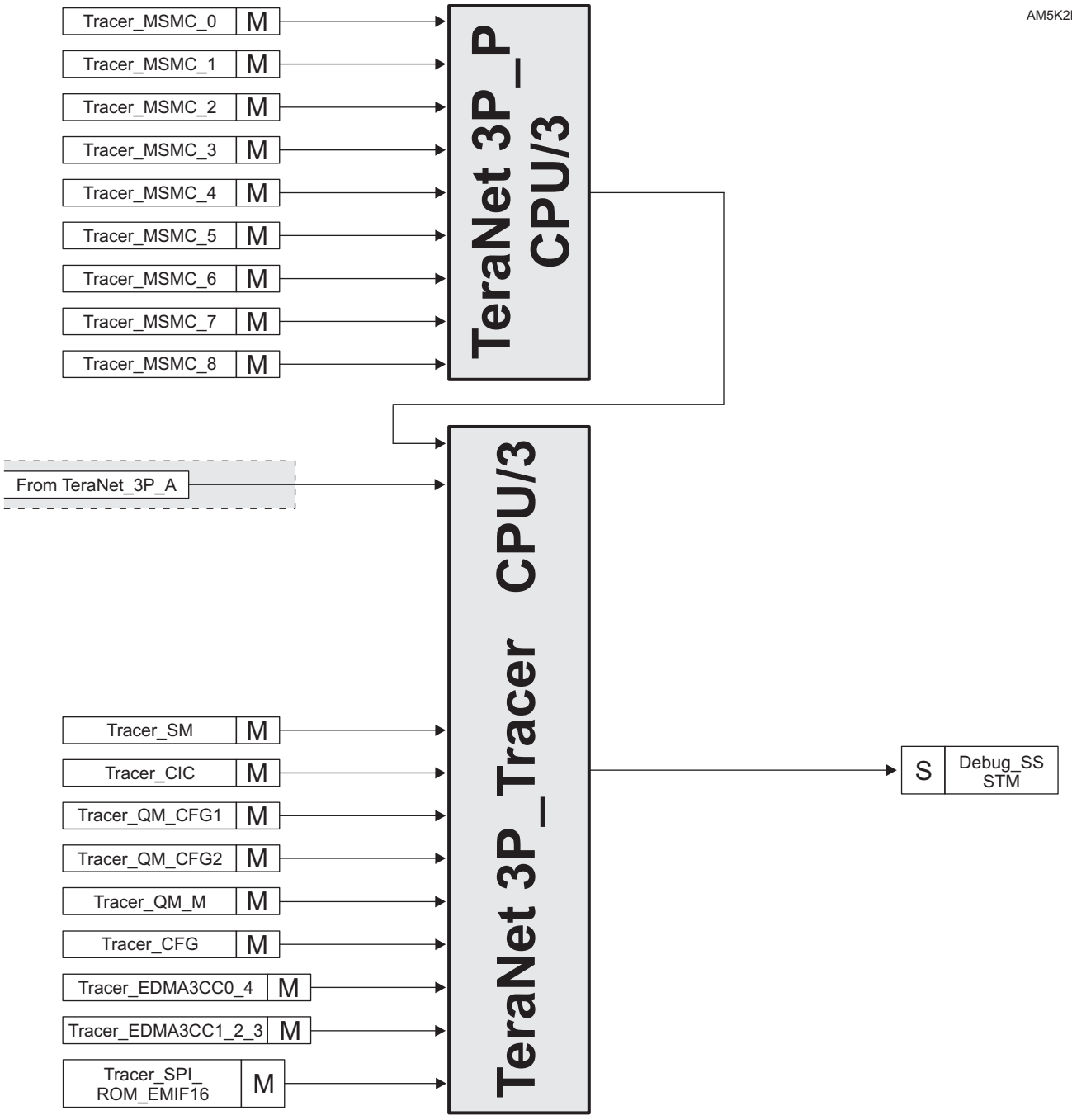


Figure 7-7. TeraNet 3P_Tracer

The following tables list the master and slave end point connections.

Intersecting cells may contain one of the following:

- **Y** — There is a connection between this master and that slave.
- **-** — There is NO connection between this master and that slave.
- ***n*** — A numeric value indicates that the path between this master and that slave goes through bridge *n*.

Table 7-2. Configuration Space Interconnect - Section 1

	SLAVES													
MASTERS	**ADTF(0-7)_CFG**	**ARM_CFG**	**BOOTCFG_CFG**	**CP_INTC_CFG**	**CPT_CFG_CFG**	**CPT_DDR3_CFG**	**CPT_INTC(0-2)_CFG**	**CPT_MSMC(0-7)_CFG**	**CPT_QM_CFG1_CFG**	**CPT_QM_CFG2_CFG**	**CPT_QM_M_CFG**	**CPT_SPI_ROM_EMIF16_CFG**	**CPT_TPCC0_4_CFG**	**CPT_TPCC1_2_3_CFG**
DBG_DAP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
TSIP_DMA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA0_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC0_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC0_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA1_TC0_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA1_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC2_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC2_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC3_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA1_TC3_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC0_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC2_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC2_WR	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC3_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC3_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_TC0_RD	13	13	13	13	13	13	13	13	13	13	13	13	13	13
EDMA3_TC0_WR	13	13	13	13	13	13	13	13	13	13	13	13	13	13
EDMA3_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC0_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC0_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC1_RD	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HyperLink0	12	12	12	12	12	12	12	12	12	12	12	12	12	12
MSMC_SYS	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
NETCP	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 7-2. Configuration Space Interconnect - Section 1 (continued)

MASTERS	SLAVES													
	ADTF(0-7)_CFG	ARM_CFG	BOOTCFG_CFG	CP_INTC_CFG	CPT_CFG_CFG	CPT_DDR3_CFG	CPT_INTC(0-2)_CFG	CPT_MSMC(0-7)_CFG	CPT_QM_CFG1_CFG	CPT_QM_CFG2_CFG	CPT_QM_M_CFG	CPT_SPI_ROM_EMIF16_CFG	CPT_TPCC0_4_CFG	CPT_TPCC1_2_3_CFG
PCIE0	12	12	12	12	12	12	12	12	12	12	12	12	12	12
PCIE1	12	12	12	12	12	12	12	12	12	12	12	12	12	12
QM_Master1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_Master2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_SEC	-	12	-	-	-	-	-	-	-	-	-	-	-	-
USB0	-	-	-	-	-	-	-	-	-	-	-	-	-	-
USB1	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 7-3. Configuration Space Interconnect - Section 2

MASTERS	SLAVES																								
	DBG_CFG	DBG_TBR_SYS	DDR3_PHY_CFG	EDMA0_CC_CFG	EDMA0_TC(0-1)_CFG	EDMA1_CC_CFG	EDMA1_TC(0-3)_CFG	EDMA2_CC_CFG	EDMA2_TC(0-3)_CFG	EDMA3_CC_CFG	EDMA3_TC(0-1)_CFG	EDMA4_CC_CFG	EDMA4_TC(0-1)_CFG	GIC_CFG	GPIO_CFG	HYPERLINK0_SERDES_CFG	I2C(0-2)_CFG	MPU(0-14)_CFG	NETCP_CFG	NETCP_SERDES_CFG	OTP_CFG	PCIE0_SERDES_CFG	PCIE1_SERDES_CFG	PLL_CTL_CFG	PSC_CFG	QM_CFG1
DBG_DAP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
TSIP_DMA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA0_CC_TR	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC0_RD	-	12	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC0_WR	-	-	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC1_RD	-	12	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC1_WR	-	-	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_CC_TR	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA1_TC0_WR	12	-	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA1_TC1_RD	-	-	-	13	13	13	13	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC1_WR	-	-	-	13	13	13	13	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC2_RD	-	-	-	14	14	14	14	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC2_WR	-	-	-	14	14	14	14	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC3_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA1_TC3_WR	12	-	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_CC_TR	-	-	-	-	-	-	-	-	Y	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC0_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC0_WR	12	-	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12

Table 7-3. Configuration Space Interconnect - Section 2 (continued)

	SLAVES																								
MASTERS	DBG_CFG	DBG_TBR_SYS	DDR3_PHY_CFG	EDMA0_CC_CFG	EDMA0_TC(0-1)_CFG	EDMA1_CC_CFG	EDMA1_TC(0-3)_CFG	EDMA2_CC_CFG	EDMA2_TC(0-3)_CFG	EDMA3_CC_CFG	EDMA3_TC(0-1)_CFG	EDMA4_CC_CFG	EDMA4_TC(0-1)_CFG	GIC_CFG	GPIO_CFG	HYPERLINK0_SERDES_CFG	I2C(0-2)_CFG	MPU(0-14)_CFG	NETCP_CFG	NETCP_SERDES_CFG	OTP_CFG	PCIE0_SERDES_CFG	PCIE1_SERDES_CFG	PLL_CTL_CFG	PSC_CFG	QM_CFG1
EDMA2_TC1_RD	-	-	-	13	13	13	13	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC1_WR	-	-	-	13	13	13	13	13	13	13	13	13	13	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC2_RD	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC2_WR	12	-	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
EDMA2_TC3_RD	-	-	-	14	14	14	14	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC3_WR	-	-	-	14	14	14	14	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_TC0_RD	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
EDMA3_TC0_WR	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
EDMA3_TC1_RD	-	14	-	14	14	14	14	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_TC1_WR	-	-	-	14	14	14	14	14	14	14	14	14	14	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC0_RD	-	12	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC0_WR	-	-	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC1_RD	-	12	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC1_WR	-	-	-	12	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-	-	-	-	-	-	-
HyperLink0_Master	12	-	12	12	12	12	12	12	12	12	12	12	12	-	12	12	12	12	12	12	12	12	12	12	12	12
MSMC_SYS	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
NETCP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCIE0	12	12	12	12	12	12	12	12	12	12	12	12	12	-	12	12	12	12	12	12	12	12	12	12	12	12
PCIE1	12	12	12	12	12	12	12	12	12	12	12	12	12	-	12	12	12	12	12	12	12	12	12	12	12	12
QM_Master1	-	-	-	12	-	12	-	12	-	12	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_Master2	-	-	-	12	-	12	-	12	-	12	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_SEC	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-	-	-	-	-
USB0	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
USB1	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 7-4. Configuration Space Interconnect - Section 3

	SLAVES																		
MASTERS	QM_CFG2	SR_CFG(0-1)	TBR_SYS_ARM	TETB0_CFG	TETB1_CFG	TETB2_CFG	TETB3_CFG	TETB4_CFG	TETB5_CFG	TETB6_CFG	TETB7_CFG	TIMER(0-19)_CFG	UART(0-1)_CFG	USB0_MMR_CFG	USB0_PHY_CFG	USB1_MMR_CFG	USB1_PHY_CFG	USIM_CFG
DBG_DAP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
TSIP_DMA	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EDMA0_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 7-4. Configuration Space Interconnect - Section 3 (continued)

MASTERS	SLAVES																	
	QM_CFG2	SR_CFG(0-1)	TBR_SYS_ARM	TETB0_CFG	TETB1_CFG	TETB2_CFG	TETB3_CFG	TETB4_CFG	TETB5_CFG	TETB6_CFG	TETB7_CFG	TIMER(0-19)_CFG	UART(0-1)_CFG	USB0_MMR_CFG	USB0_PHY_CFG	USB1_MMR_CFG	USB1_PHY_CFG	USIM_CFG
EDMA0_TC0_RD	-	-	-	-	-	-	-	12	12	-	-	-	-	-	-	-	-	-
EDMA0_TC0_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA0_TC1_RD	-	-	-	-	-	-	-	12	12	-	-	-	-	-	-	-	-	-
EDMA0_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC0_RD	12	12	12	-	-	-	-	12	12	-	-	12	12	12	12	12	12	12
EDMA1_TC0_WR	12	12	12	-	-	-	-	-	-	-	-	12	12	12	12	12	12	12
EDMA1_TC1_RD	-	-	-	13	13	-	-	-	-	13	-	-	-	-	-	-	-	-
EDMA1_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC2_RD	-	-	-	-	-	14	14	-	-	-	14	-	-	-	-	-	-	-
EDMA1_TC2_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA1_TC3_RD	12	12	12	-	-	-	-	12	12	-	-	12	12	12	12	12	12	12
EDMA1_TC3_WR	12	12	12	-	-	-	-	-	-	-	-	12	12	12	12	12	12	12
EDMA2_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC0_RD	12	12	12	-	-	-	-	Y	Y	-	-	12	12	12	12	12	12	12
EDMA2_TC0_WR	12	12	12	-	-	-	-	-	-	-	-	12	12	12	12	12	12	12
EDMA2_TC1_RD	-	-	-	13	13	-	-	-	-	13	-	-	-	-	-	-	-	-
EDMA2_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA2_TC2_RD	12	12	12	-	-	-	-	12	12	-	-	12	12	12	12	12	12	12
EDMA2_TC2_WR	12	12	12	-	-	-	-	-	-	-	-	12	12	12	12	12	12	12
EDMA2_TC3_RD	-	-	-	-	-	14	14	-	-	-	14	-	-	-	-	-	-	-
EDMA2_TC3_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA3_TC0_RD	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
EDMA3_TC0_WR	13	13	13	-	-	-	-	-	-	-	-	13	13	13	13	13	13	13
EDMA3_TC1_RD	-	-	-	14	14	14	14	14	14	14	14	-	-	-	-	-	-	-
EDMA3_TC1_WR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_CC_TR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC0_RD	-	-	12	-	-	-	-	12	12	-	-	-	-	-	-	-	-	-
EDMA4_TC0_WR	-	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EDMA4_TC1_RD	-	-	12	-	-	-	-	12	12	-	-	-	-	-	-	-	-	-
EDMA4_TC1_WR	-	-	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HyperLink0_Master	12	12	12	-	-	-	-	-	-	-	-	12	12	12	12	12	12	12
MSMC_SYS	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
NETCP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCIE0	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
PCIE1	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
QM_Master1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_Master2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QM_SEC	-	-	12	-	-	-	-	-	-	-	-	-	-	12	-	12	-	-
USB0	-	-	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-
USB1	-	-	12	12	12	12	12	12	12	12	12	-	-	-	-	-	-	-

7.4 Bus Priorities

The priority level of all master peripheral traffic is defined at the TeraNet boundary. User-programmable priority registers allow software configuration of the data traffic through the TeraNet. Note that a lower number means higher priority — PRI = 000b = urgent, PRI = 111b = low.

All other masters provide their priority directly and do not need a default priority setting. All the Packet DMA-based peripherals also have internal registers to define the priority level of their initiated transactions.

The Packet DMA secondary port is one master port that does not have priority allocation register inside the Multicore Navigator. The priority level for transaction from this master port is described by the QM_PRIORITY bit field in the CHIP_MISC_CTL0 register shown in and Table 8-48.

For all other modules, see the respective User's Guides listed in Section 3.4 for programmable priority registers.

8 Device Boot and Configuration

8.1 Device Boot

8.1.1 Boot Sequence

The boot sequence is a process by which the internal memory is loaded with program and data sections. The boot sequence is started automatically after each power-on reset or warm reset.

The

AM5K2E0x supports several boot processes that begins execution at the ROM base address, which contains the bootloader code necessary to support various device boot modes. The boot processes are software-driven and use the BOOTMODE[15:0] device configuration inputs to determine the software configuration that must be completed. For more details on boot sequence see the *KeyStone II Architecture ARM Bootloader User's Guide* (SPRUHJ3).

For AM5K2E0x non-secure devices, there is only one type of booting: the ARM CorePac as the boot master.The ARM CorePac does not support no-boot mode. The ARM CorePac needs to read the bootmode register to determine how to proceed with the boot.

Table 8-1 shows addresses reserved for boot by the ARM CorePac.

Table 8-1. ARM Boot RAM Memory Map

START ADDRESS	SIZE	DESCRIPTION
0xc17_e000	0xc00	Context RAM not Scrubbed on Secure boot
0xc18_6f80	0x80	Global Level 0 Non-secure Translation table
0xc18_7000	0x5000	Global Non-secure Page Table for memory Covering ROM
0xc18_c000	0x1000	Core 0 Non-secure Level 1 Translation table
0xc18_d000	0x1000	Core 1 Non-secure Level 1 Translation table
0xc18_e000	0x1000	Core 2 Non-secure Level 1 Translation table
0xc18_f000	0x1000	Core 3 Non-secure Level 1 Translation table
0xc19_0000	0x7e80	Packet Memory Buffer
0xc19_7e80	080	PCIE Block
0xc19_7f00	4	Host Data Address (boot magic address for secure boot through master peripherals)
0xc1a_6e00	0x200	DDR3 Configuration Structure
0xc1a_7000	0x3000	Boot Data
0xc1a_a000	0x3000	Supervisor Stack, Each Core Gets 0xc000 Bytes
0xc1a_d000	4	ARM Boot Magic Address, Core 0
0xc1a_d004	4	ARM Boot Magic Address, Core 1
0xc1a_d008	4	ARM Boot Magic Address, Core 2
0xc1a_d00c	4	ARM Boot Magic Address, Core 3
0xc1a_e000	0x400	Abort Stack, Core 0
0xc1a_e400	0x400	Abort Stack, Core 1
0xc1a_e800	0x400	Abort Stack, Core 2
0xc1a_ec00	0x400	Abort Stack, Core 3
0xc1a_f000	0x400	Unknown Mode Stack, Core 0
0xc1a_f400	0x400	Unknown mOde Stack, Core 1
0xc1a_f800	0x400	Unknown Mode Stack, Core 2
0xc1a_fc00	0x400	Unknown Mode Stack, Core 3
0xc1b_0000	0x180	Boot Version String, Core 0
0xc1b_0180	0x80	Boot Status Stack, Core 0
0xc1b_0200	0x100	Boot Stats, Core 0
0xc1b_0300	0x100	Boot Log, Core 0

Table 8-1. ARM Boot RAM Memory Map (continued)

START ADDRESS	SIZE	DESCRIPTION
0xc1b_0400	0x100	Boot RAM Call Table, Core 0
0xc1b_0500	0x100	Boot Parameter Tables, Core 0
0xc1b_0600	0x19e0	Boot Data, Core 0
0xc1b_1fe0	0x1010	Boot Trace, Core 0
0xc1b_4000	0x180	Boot Version String, Core 1
0xc1b_4180	0x80	Boot Status Stack, Core 1
0xc1b_4200	0x100	Boot Stats, Core 1
0xc1b_4300	0x100	Boot Log, Core 1
0xc1b_4400	0x100	Boot RAM Call Table, Core 1
0xc1b_4500	0x100	Boot Parameter Tables, Core 1
0xc1b_4600	0x19e0	Boot Data, Core 1
0xc1b_5fe0	0x1010	Boot Trace, Core 1
0xc1b_6000	0x180	Boot Version String, Core 2
0xc1b_6180	0x80	Boot Status Stack, Core 2
0xc1b_6200	0x100	Boot Stats, Core 2
0xc1b_6300	0x100	Boot Log, Core 2
0xc1b_6400	0x100	Boot RAM Call Table, Core 2
0xc1b_6500	0x100	Boot Parameter Tables, Core 2
0xc1b_6600	0x19e0	Boot Data, Core 2
0xc1b_7fe0	0x1010	Boot Trace, Core 2
0xc1b_8000	0x180	Boot Version String, Core 3
0xc1b_8180	0x80	Boot Status Stack, Core 3
0xc1b_8200	0x100	Boot Stats, Core 3
0xc1b_8300	0x100	Boot Log, Core 3
0xc1b_8400	0x100	Boot RAM Call Table, Core 3
0xc1b_8500	0x100	Boot Parameter Tables, Core 3
0xc1b_8600	0x19e0	Boot Data, Core 3
0xc1b_9fe0	0x1010	Boot Trace, Core 3
0xc1c_0000	0x4_0000	Secure MSMC

8.1.2 Boot Modes Supported

The device supports several boot processes, which leverage the internal boot ROM. Most boot processes are software-driven, using the BOOTMODE[15:0] device configuration inputs to determine the software configuration that must be completed. From a hardware perspective, there are two possible boot modes:

- **Public ROM Boot when the ARM CorePac Core0 is the boot master** — In this boot mode, the ARM CorePac performs the boot process. When the ARM CorePac Core0 finishes the boot process, it may send Cortex-A15 processor cores through IPC registers.
- **Secure ROM Boot when the ARM CorePac0 is the boot master** — The ARM CorePac Core0 are released from reset simultaneously and begin executing from secure ROM. The ARM CorePac Core0 initiates the boot process. For more information, refer to the Secure device Addendum.

The boot process performed by the ARM CorePac Core0 in public ROM boot and secure ROM boot are determined by the BOOTMODE[15:0] value in the DEVSTAT register. The ARM CorePac Core0 read this value, and then execute the associated boot process in software. The figure below shows the bits associated with BOOTMODE[15:0] pins (DEVSTAT[16:1] register bits) when the ARM CorePac is the boot master. Note that Figure 8-1 does not include bit 0 of the DEVSTAT contents. Bit 0 is used to select overall system endianess that is independent of the boot mode.

The boot ROM will continue attempting to boot in this mode until successful or an unrecoverable error occurs.

The PLL settings are shown at the end of this section, and the PLL set-up details can be found in Section 10.5.

NOTE

It is important to keep in mind that BOOTMODE[15:0] **pins** map to DEVSTAT[16:1] **bits** of the DEVSTAT register.

Figure 8-1. DEVSTAT Boot Mode Pins ROM Mapping

<table>
<tr><th colspan="17">DEVSTAT Boot Mode Pins ROM Mapping</th></tr>
<tr><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>Mode</th></tr>
<tr><td>X</td><td>X</td><td>0</td><td>X</td><td>PLLEN</td><td colspan="3" rowspan="2">X</td><td rowspan="10">0 (ARM Boot Master)</td><td colspan="3" rowspan="2">SYS PLL CONFIG</td><td rowspan="3">Min</td><td>0</td><td>0</td><td>0</td><td>SLEEP</td></tr>
<tr><td colspan="2">SlaveAddr</td><td>1</td><td colspan="2">Port</td><td>0</td><td>0</td><td>0</td><td>I2C SLAVE</td></tr>
<tr><td colspan="3">X</td><td colspan="2">Bus Addr</td><td colspan="3">Param Idx</td><td>X</td><td>Port</td><td>0</td><td>0</td><td>1</td><td>I2C MASTER</td></tr>
<tr><td colspan="2">Width</td><td colspan="2">Csel</td><td colspan="2">Mode</td><td colspan="2">Port</td><td colspan="3">Param Idx</td><td>0</td><td>1</td><td>0</td><td>SPI</td></tr>
<tr><td>0</td><td colspan="2">Base Addr</td><td rowspan="2">Wait</td><td>Width</td><td rowspan="3">X</td><td colspan="2" rowspan="2">Chip Sel</td><td rowspan="5">SYS PLL CONFIG</td><td>0</td><td>0</td><td>1</td><td>1</td><td>EMIF</td></tr>
<tr><td>1</td><td colspan="2">First Block</td><td>Clear</td><td>Min</td><td>0</td><td>1</td><td>1</td><td>NAND</td></tr>
<tr><td>NETCP clk</td><td colspan="2">Ref clk</td><td colspan="2">Ext Con</td><td colspan="2">Lane Setup</td><td></td><td>1</td><td>0</td><td>1</td><td>Ethernet</td></tr>
<tr><td>X</td><td colspan="4">Bar Config</td><td>Port</td><td colspan="2">X</td><td>0</td><td>1</td><td>1</td><td>0</td><td>PCIe</td></tr>
<tr><td>Port</td><td colspan="2">Ref clk</td><td colspan="2">Data Rate</td><td colspan="3" rowspan="2">X</td><td>1</td><td>1</td><td>1</td><td>0</td><td>HyperLink</td></tr>
<tr><td colspan="4">X</td><td>Port</td><td>Min</td><td>1</td><td>1</td><td>1</td><td>UART</td></tr>
</table>

8.1.2.1 Boot Device Field

The Boot Device field DEVSTAT[16-14-4-3-2-1] defines the boot device that is chosen. Table 8-2 shows the supported boot modes.

Table 8-2. Boot Mode Pins: Boot Device Values

Bit	Field	Description
16, 14, 4, 3, 2, 1	Boot Device	Device boot mode - ARM is a boot master when BOOTMODE[8]=0 • Sleep = X0[Min]000b • I2C Slave = [Slave Addr1]1[Min]000 b • I2C Master = XX[Min]001b • SPI = [Width][Csel0][Min]010b • EMIF = 0X0011b • NAND = 1X[Min]011b • Ethernet (SGMII) = [Pa clk][Ref Clk0][Min]101b • PCI = XBar Config2]0110b • Hyperlink = [Port][Ref Clk0]1110b • UART = XX[Min]111b

8.1.2.2 Device Configuration Field

The device configuration fields DEVSTAT[16:1] are used to configure the boot peripheral and, therefore, the bit definitions depend on the boot mode.

8.1.2.2.1 Sleep Boot Mode Configuration

Figure 8-2. Sleep Boot Mode Configuration Fields Description

DEVSTAT Boot Mode Pins ROM Mapping																
16	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X	X	0	X	PLLEN	X			Boot Master	Sys PLL Config			Min	000			Lendian

Table 8-3. Sleep Boot Configuration Field Descriptions

Bit	Field	Description
16-15	Reserved	Reserved
14	Boot Devices	Boot Device- used in conjunction with Boot Devices [Used in conjunction with bits 3-1] • 0 = Sleep (default) • Others = Other boot modes
13	Reserved	
12	PLLEN	Enable the System PLL • 0 = PLL disabled (default) • 1 = PLL enabled
11-9	Reserved	Reserved
8	Boot Master	This pin must be pulled down to GND
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits. This will set the PLL to the maximum clock setting for the device. Table 8-24 shows settings for various input clock frequencies.
4	Min	Minimum boot configuration select bit. • 0 = Minimum boot pin select disabled • 1 = Minimum boot pin select enabled. When Min = 1, a predetermined set of values is configured (see the Device Configuration Field Descriptions table for configuration bits with a "(default)" tag added in the description column). When Min = 0, all fields must be independently configured.
3-1	Boot Devices	Boot Devices[3:1] used in conjunction with Boot Device [14] • 000 = Sleep • Others = Other boot modes
0	Lendian	Endianess (device) • 0 = Big endian • 1 = Little endian

8.1.2.2.2 I²C Boot Device Configuration

8.1.2.2.2.1 I²C Passive Mode

In passive mode, the device does not drive the clock, but simply acks data received on the specified address.

Figure 8-3. I²C Passive Mode Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping																
16	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Slave Addr		1	Port		X			Boot Master	Sys PLL Config			Min	000			Lendian

Table 8-4. I²C Passive Mode Device Configuration Field Descriptions

Bit	Field	Description
16-15	Slave Addr	I²C Slave boot bus address • 0 = I²C slave boot bus address is 0x00 • 1 = I²C slave boot bus address is 0x10 (default) • 2 = I²C slave boot bus address is 0x20 • 3 = I²C slave boot bus address is 0x30

Table 8-4. I²C Passive Mode Device Configuration Field Descriptions (continued)

Bit	Field	Description
14	Boot Devices	Boot Device[14] used in conjunction with Boot Devices [Use din conjunction with bits 3-1] • 0 = Other boot modes • 1= I²C Slave boot mode
13-12	Port	I²C port number • 0 = I²C0 • 1 = I²C1 • 2 = I²C2 • 3 = Reserved
11-9	Reserved	Reserved
8	Boot Master	This pin must be pulled down to GND
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits. This will set the PLL to the maximum clock setting for the device. Table 8-24 shows settings for various input clock frequencies.
4	Min	Minimum boot configuration select bit. • 0 = Minimum boot pin select disabled • 1 = Minimum boot pin select enabled. When Min = 1, a predetermined set of values is configured (see the Device Configuration Field Descriptions table for configuration bits with a "(default)" tag added in the description column). When Min = 0, all fields must be independently configured.
3-1	Boot Devices	Boot Devices[3:1] used in conjunction with Boot Device [14] • 000 = I²C Slave • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.2.2 I²C Master Mode

In master mode, the I²C device configuration uses ten bits of device configuration instead of seven as used in other boot modes. In this mode, the device makes the initial read of the I²C EEPROM while the PLL is in bypass mode. The initial read contains the desired clock multiplier, which must be set up prior to any subsequent reads.

Figure 8-4. I²C Master Mode Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping																
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			Bus Addr		Param Idx/Offset			Boot Master	Reserved	Port		Min	001			Lendian

Table 8-5. I²C Master Mode Device Configuration Field Descriptions

Bit	Field	Description
16-14	Reserved	Reserved
13-12	Bus Addr	I²C bus address slave device • 0 = I²C slave boot bus address is 0x50 (default) • 1 = I²C slave boot bus address is 0x51 • 2 = I²C slave boot bus address is 0x52 • 3 = I²C slave boot bus address is 0x53
11-9	Param Idx	Parameter Table Index • 0-7 = This value specifies the parameter table index (default = 0)
8	Boot Master	This pin must be pulled down to GND
7	Reserved	Reserved

Table 8-5. I²C Master Mode Device Configuration Field Descriptions (continued)

Bit	Field	Description
6-5	Port	I²C port number • 0 = I²C0 (default) • 1 = I²C1 • 2 = I²C2 • 3 = Reserved
4	Min	Minimum boot configuration select bit. • 0 = Minimum boot pin select disabled • 1 = Minimum boot pin select enabled. When Min = 1, a predetermined set of values is configured (see the Device Configuration Field Descriptions table for configuration bits with a "(default)" tag added in the description column). When Min = 0, all fields must be independently configured.
3-1	Boot Devices	Boot Devices[3:1] • 001 = I²C Master • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.2.3 SPI Boot Device Configuration

Figure 8-5. SPI Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Width		Csel		Mode		Port		Boot Master	Param Ind			Min	010			Lendian

Table 8-6. SPI Device Configuration Field Descriptions

Bit	Field	Description
16-15	Width	SPI address width configuration • 0 = 16-bit address values are used • 1 = 24-bit address values are used (default)
14-13	Csel	The chip select field value 0-3 (default = 0)
12-11	Mode	Clk Polarity/ Phase • 0 = Data is output on the rising edge of SPICLK. Input data is latched on the falling edge. • 1 = Data is output one half-cycle before the first rising edge of SPICLK and on subsequent falling edges. Input data is latched on the rising edge of SPICLK. • 2 = Data is output on the falling edge of SPICLK. Input data is latched on the rising edge (default). • 3 = Data is output one half-cycle before the first falling edge of SPICLK and on subsequent rising edges. Input data is latched on the falling edge of SPICLK.
10-9	Port	Specify SPI port • 0 = SPI0 used (default) • 1 = SPI1 used • 2 = SPI2 used • 3 = Reserved
8	Boot Master	This pin must be pulled down to GND
7-5	Param Idx	Parameter Table Index • 0-7 = This value specifies the parameter table index (default = 0)
4	Min	Minimum boot configuration select bit. • 0 = Minimum boot pin select disabled • 1 = Minimum boot pin select enabled. When Min = 1, a predetermined set of values is configured (see the Device Configuration Field Descriptions table for configuration bits with a "(default)" tag added in the description column). When Min = 0, all fields must be independently configured.
3-1	Boot Devices	Boot Devices[3:1] • 010 = SPI boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.2.4 EMIF Boot Device Configuration

Figure 8-6. EMIF Boot Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping																
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Base Addr		Wait	Width	X	Chip Sel		Boot Master	Sys PLL Cfg			0	011			Lendian

Table 8-7. EMIF Boot Device Configuration Field Descriptions

Bit	Field	Description
16	Boot Devices	Boot Devices[16] used conjunction with Boot Devices[4] and Boot Devices [Used in conjunction with bits 3-1] • 0 = EMIF boot mode • 1 = Other boot modes
15-14	Base Addr	Base address (0-3) used to calculate the branch address. Branch address is the chip select plus Base Address *16MB
13	Wait	Extended Wait • 0 = Extended Wait disabled • 1 = Extended Wait enabled
12	Width	EMIF Width • 0 = 8-bit EMIF Width • 1 = 16-bit EMIF Width
11	Reserved	Reserved
10-9	Chip Sel	Chip Sel that specifies the chip select region, EMIF16 CS2-EMIF16 CS5. • 00 = EMIF16 CS2($\overline{\text{EMIFCE0}}$) • 01 = EMIF16 CS3 ($\overline{\text{EMIFCE1}}$) • 10 = EMIF16 CS4 ($\overline{\text{EMIFCE2}}$) • 11 = EMIF16 CS5 ($\overline{\text{EMIFCE3}}$)
8	Boot Master	This pin must be pulled down to GND
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits. This will set the PLL to the maximum clock setting for the device. Table 8-24 shows settings for various input clock frequencies.
4-1	Boot Devices	Boot Devices[4] used conjunction with Boot Devices[16] • 0011 = EMIF boot mode • 1XXX = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.2.5 NAND Boot Device Configuration

Figure 8-7. NAND Boot Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping																
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	First Block			Clear	X	Chip Sel		Boot Master	Sys PLL Cfg			Min	011			Lendian

Table 8-8. NAND Boot Device Configuration Field Descriptions

Bit	Field	Description
16	Boot Devices	Boot Devices[16] used conjunction with Boot Devices [3-1] • 0 = Other boot modes • 1 = NAND boot mode
15-13	First Block	First Block. This value is used to calculate the first block read. The first block read is the first block value *16.
12	Clear	ClearNAND • 0 = Device is not a ClearNAND (default) • 1 = Device is a ClearNAND
11-9	Chip Sel	Chip Sel that specifies the chip select region, EMIF16 CS2-EMIF16 CS5. • 00 = EMIF16 CS2($\overline{\text{EMIFCE0}}$) • 01 = EMIF16 CS3 ($\overline{\text{EMIFCE1}}$) • 10 = EMIF16 CS4 ($\overline{\text{EMIFCE2}}$) • 11 = EMIF16 CS5 ($\overline{\text{EMIFCE3}}$)
8	Boot Master	This pin must be pulled down to GND

Table 8-8. NAND Boot Device Configuration Field Descriptions (continued)

Bit	Field	Description
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits. This will set the PLL to the maximum clock setting for the device. Table 8-24 shows settings for various input clock frequencies.
4	Min	Minimum boot pin select. When Min is 1, it means that the BOOTMODE [15:3] pins are don't cares. Only BOOTMODE [2:0] pins (DEVSTAT[3:1]) will determine boot. Default values are assigned to values that would normally be set by the other BOOTMODE pins when Min is 0. • 0 = Minimum boot pin select disabled • 1 = Minimum boot pin select enabled.
3-1	Boot Devices	Boot Devices • 011 = NAND boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.3 Ethernet (SGMII) Boot Device Configuration

Figure 8-8. Ethernet (SGMII) Boot Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NETCP clk	Ref Clock		Ext Con		X	Lane Setup		Boot Master	Sys PLL Cfg			Min	101			Lendian

Table 8-9. Ethernet (SGMII) Boot Device Configuration Field Descriptions

Bit	Field	Description
16	NETCP clk	NETCP clock reference • 0 = NETCP clocked at the same reference as the core reference • 1 = NETCP clocked at the same reference as the SerDes reference (default)
15-14	Ref Clock	Reference clock frequency • 0 = 125MHz • 1 = 156.25MHz (default) • 2 = Reserved • 3 = Reserved
13-12	Ext Con	External connection mode • 0 = MAC to MAC connection, master with auto negotiation • 1 = MAC to MAC connection, slave with auto negotiation (default) • 2 = MAC to MAC, forced link, maximum speed • 3 = MAC to fiber connection
11-9	Lane Setup	Lane Setup. • 0 = All SGMII ports enabled (default) • 1 = Only SGMII port 0 enabled • 2 = SGMII port 0 and 1 enabled • 3 = SGMII port 0, 1 and 2 enabled • 4-5 = Reserved
8	Boot Master	This pin must be pulled down to GND
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits. This will set the PLL to the maximum clock setting for the device. Default system reference clock is 156.25 MHz. Table 8-24 shows settings for various input clock frequencies. (default = 4)

Table 8-9. Ethernet (SGMII) Boot Device Configuration Field Descriptions (continued)

Bit	Field	Description
4	Min	Minimum boot configuration select bit. • 0 = Minimum boot pin select disabled • 1 = Minimum boot pin select enabled. When Min = 1, a predetermined set of values is configured (see the Device Configuration Field Descriptions table for configuration bits with a "(default)" tag added in the description column). When Min = 0, all fields must be independently configured.
3-1	Boot Devices	Boot Devices • 101 = Ethernet boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.3.1 PCIe Boot Device Configuration

Figure 8-9. PCIe Boot Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping															
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ref clk	Bar Config				Port	X		Boot Master	Sys PLL Cfg			0110				Lendian

Table 8-10. PCIe Boot Device Configuration Field Descriptions

Bit	Field	Description
16	Ref clk	PCIe Reference clock frequency • 0 = 100MHz • 1 = Reserved
15-12	Bar Config	PCIe BAR registers configuration This value can range from 0 to 0xf. See Table 8-11.
11	Port	PCIe Port number (0-1)
10-9	Reserved	
8	Boot Master	This pin must be pulled down to GND
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits.This will set the PLL to the maximum clock setting for the device. Default system reference clock is 156.25 MHz. Table 8-24 shows settings for various input clock frequencies.
4-1	Boot Devices	Boot Devices[4:1] • 0110 = PCIe boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

Table 8-11. BAR Config / PCIe Window Sizes

BAR CFG	BAR0	32-BIT ADDRESS TRANSLATION					64-BIT ADDRESS TRANSLATION	
		BAR1	BAR2	BAR3	BAR4	BAR5	BAR2/3	BAR4/5
0b0000	PCIe MMRs	32	32	32	32	Clone of BAR4		
0b0001		16	16	32	64			
0b0010		16	32	32	64			
0b0011		32	32	32	64			
0b0100		16	16	64	64			
0b0101		16	32	64	64			
0b0110		32	32	64	64			
0b0111		32	32	64	128			
0b1000		64	64	128	256			
0b1001		4	128	128	128			
0b1010		4	128	128	256			
0b1011		4	128	256	256			
0b1100							256	256
0b1101							512	512
0b1110							1024	1024
0b1111							2048	2048

8.1.2.3.2 HyperLink Boot Device Configuration

Figure 8-10. HyperLink Boot Device Configuration Fields

DEVSTAT Boot Mode Pins ROM Mapping																
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	RefClk		Data Rate		X			Boot Master	Sys PLL Cfg			1110				Lendian

Table 8-12. HyperLink Boot Device Configuration Field Descriptions

Bit	Field	Description
16	Reserve	
15-14	Ref Clocks	HyperLink reference clock configuration • 0 = 125 MHz • 1 = 156.25 MHz • 2-3 = Reserved
13-12	Data Rate	HyperLink data rate configuration • 0 = 1.25 GBs • 1 = 3.125 GBs • 2 = 6.25 GBs • 3 = 12.5GBs
11-9	Reserved	
8	Boot Master	This pin must be pulled down to GND
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits. This will set the PLL to the maximum clock setting for the device. Default system reference clock is 156.25 MHz. Table 8-24 shows settings for various input clock frequencies.
4-1	Boot Devices	Boot Devices[4:1] • 1110 = HyperLink boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.3.3 UART Boot Device Configuration

Figure 8-11. UART Boot Mode Configuration Field Description

DEVSTAT Boot Mode Pins ROM Mapping																
16	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X	X	X	X	Port	X	X	X	Boot Master	Sys PLL Config			Min	111			Lendian

Table 8-13. UART Boot Configuration Field Descriptions

Bit	Field	Description
16-13	Reserved	Not Used
12	Port	UART Port number • 0 = UART0 • 1 = UART1
11-9	Reserved	Not Used
8	Boot Master	This pin must be pulled down to GND
7-5	SYS PLL Setting	The PLL default settings are determined by the [7:5] bits. This will set the PLL to the maximum clock setting for the device. Table 8-24 shows settings for various input clock frequencies. (default = 4)
4	Min	Minimum boot configuration select bit. • 0 = Minimum boot pin select disabled • 1 = Minimum boot pin select enabled. When Min = 1, a predetermined set of values is configured (see the Device Configuration Field Descriptions table for configuration bits with a "(default)" tag added in the description column). When Min = 0, all fields must be independently configured.
3-1	Boot Devices	Boot Devices[3:1] • 111 = UART boot mode • Others = Other boot modes
0	Lendian	Endianess • 0 = Big endian • 1 = Little endian

8.1.2.4 Boot Parameter Table

The ROM Bootloader (RBL) uses a set of tables to carry out the boot process. The boot parameter table is the most common format the RBL employs to determine the boot flow. These boot parameter tables have certain parameters common across all the boot modes, while the rest of the parameters are unique to the boot modes. The common entries in the boot parameter table are shown in Table 8-14.

Table 8-14. Boot Parameter Table Common Parameters

BYTE OFFSET	NAME	DESCRIPTION
0	Length	The length of the table, including the length field, in bytes.
2	Checksum	The 16 bits ones complement of the ones complement of the entire table. A value of 0 will disable checksum verification of the table by the boot ROM.
4	Boot Mode	Internal values used by RBL for different boot modes.
6	Port Num	Identifies the device port number to boot from, if applicable
8	SW PLL, MSW	PLL configuration, MSW
10	SW PLL, LSW	PLL configuration, LSW
12	Reserved	Reserved
14	Reserved	Reserved
16	System Freq	The Frequency of the system clock in MHz
18	Core Freq	The frequency of the core clock in MHz
20	Boot Master	Set to FALSE if ARM is the master core.

8.1.2.4.1 EMIF16 Boot Parameter Table

Table 8-15. EMIF16 Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
22	Options	Async Config Parameters are used. • 0 = Value in the async config paramters are not used to program async config registers. • 1 = Value in the async config paramters are used to program async config registers.	NO
24	Type	Set to 0 for EMIF16 (NOR) boot	NO
26	Branch Address MSW	Most significant bit for Branch address (depends on chip select)	YES
28	Branch Address LSW	Least significant bit for Branch address (depends on chip select)	YES
30	Chip Select	Chip Select for the NOR flash	YES
32	Memory Width	Memory width of the EMIF16 bus (16 bits)	YES
34	Wait Enable	Extended wait mode enabled • 0 = Wait enable is disabled • 1 = Wait enable is enabled	YES
36	Async Config MSW	Async Config Register MSW	NO
38	Async Config LSW	Async Config Register LSW	NO

8.1.2.4.2 Ethernet Boot Parameter Table

Table 8-16. Ethernet Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
22	Options	Bits 02 - 00 Interface • 000 - 100 = Reserved • 101 = SGMII • 110 = Reserved • 111 = Reserved Bits 03 HD • 0 = Half Duplex • 1 = Full Duplex Bit 4 Skip TX • 0 = Send Ethernet Ready Frame every 3 seconds • 1 = Don't send Ethernet Ready Frame Bits 06 - 05 Initialize Config • 00 = Switch, SerDes, SGMII and NETCP are configured • 01 = Initialization is not done for the peripherals that are already enabled and running. • 10 = Reserved • 11 = None of the Ethernet system is configured. Bits 15 - 07 Reserved	NO
24	MAC High	The 16 MSBs of the MAC address to receive during boot	NO
26	MAC Med	The 16 middle bits of the MAC address to receive during boot	NO
28	MAC Low	The 16 LSBs of the MAC address to receive during boot	NO
30	Multi MAC High	The 16 MSBs of the multi-cast MAC address to receive during boot	NO
32	Multi MAC Med	The 16 middle bits of the multi-cast MAC address to receive during boot	NO
34	Multi MAC Low	The 16 LSBs of the multi-cast MAC address to receive during boot	NO

Table 8-16. Ethernet Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
36	Source Port	The source UDP port to accept boot packets from. A value of 0 will accept packets from any UDP port	NO
38	Dest Port	The destination port to accept boot packets on.	NO
40	Device ID 12	The first two bytes of the device ID. This is typically a string value, and is sent in the Ethernet ready frame	NO
42	Device ID 34	The 2nd two bytes of the device ID.	NO
44	Dest MAC High	The 16 MSBs of the MAC destination address used for the Ethernet ready frame. Default is broadcast.	NO
46	Dest MAC Med	The 16 middle bits of the MAC destination address	NO
48	Dest MAC Low	The 16 LSBs of the MAC destination address	NO
50	Lane Enable	One bit per lane. • 0 - Lane disabled • 1 - Lane enabled	
52	SGMII Config	Bits 0-3 are the config index, bit 4 set if direct config used, bit 5 set if no configuration done	NO
54	SGMII Control	The SGMII control register value	NO
56	SGMII Adv Ability	The SGMII ADV Ability register value	NO
58	SGMII TX Cfg High	The 16 MSBs of the SGMII Tx config register	NO
60	SGMII TX Cfg Low	The 16 LSBs of the SGMII Tx config register	NO
62	SGMII RX Cfg High	The 16 MSBs of the SGMII Rx config register	NO
64	SGMII RX Cfg Low	The 16 LSBs of the SGMII Rx config register	NO
66	SGMII Aux Cfg High	The 16 MSBs of the SGMII Aux config register	NO
68	SGMII Aux Cfg Low	The 16 LSBs of the SGMII Aux config register	NO
70	PKT PLL Cfg MSW	The packet subsystem PLL configuration, MSW	NO
72	PKT PLL CFG LSW	The packet subsystem PLL configuration, LSW	NO

8.1.2.4.3 PCIe Boot Parameter Table

Table 8-17. PCIe Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
22	Options	Bits 00 Mode • 0 = Host Mode (Direct boot mode) • 1 = Boot Table Boot Mode Bits 01 Configuration of PCIe • 0 = PCIe is configured by RBL • 1 = PCIe is not configured by RBL Bit 03-02 Reserved Bits 04 Multiplier • 0 = SERDES PLL configuration is done based on SERDES register values • 1 = SERDES PLL configuration based on the reference clock values Bits 05-15 Reserved	NO
24	Address Width	PCI address width, can be 32 or 64	YES with in conjunction with BAR sizes
26	Link Rate	SerDes frequency, in Mbps. Can be 2500 or 5000	NO

Table 8-17. PCIe Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
28	Reference clock	Reference clock frequency, in units of 10 kHz. Value values are 10000 (100 MHz), 12500 (125 MHz), 15625 (156.25 MHz), 25000 (250 MHz) and 31250 (312.5 MHz). A value of 0 means that value is already in the SerDes cfg parameters and will not be computed by the boot ROM.	NO
30	Window 1 Size	Window 1size.	YES
32	Window 2 Size	Window 2 size.	YES
34	Window 3 Size	Window 3 size. Valid only if address width is 32.	YES
36	Window 4 Size	Window 4 Size. Valid only if the address width is 32.	YES
38	Vendor ID	Vendor ID	NO
40	Device ID	Device ID	NO
42	Class code Rev ID MSW	Class code revision ID MSW	NO
44	Class code Rev ID LSW	Class code revision ID LSW	NO
46	SerDes cfg msw	PCIe SerDes config word, MSW	NO
48	SerDes cfg lsw	PCIe SerDes config word, LSW	NO
50	SerDes lane 0 cfg msw	SerDes lane config word, msw lane 0	NO
52	SerDes lane 0 cfg lsw	SerDes lane config word, lsw, lane 0	NO
54	SerDes lane 1 cfg msw	SerDes lane config word, msw, lane 1	NO
56	SerDes lane 1 cfg lsw	SerDes lane config word, lsw, lane 1	NO
58	Timeout period (Secs)	The timeout period. Values 0 disables the time out	

8.1.2.4.4 I^2C Boot Parameter Table

Table 8-18. I^2C Boot Parameter Table

OFFSET	FIELD	VALUE	CONFIGURED THROUGH BOOT CONFIGURATION PINS
22	Option	Bits 02 - 00 Mode • 000 = Boot Parameter Table Mode • 001 = Boot Table Mode • 010 = Boot Config Mode • 011 = Load GP header format data • 100 = Slave Receive Boot Config Bits 15 - 03= Reserved	NO
24	Boot Dev Addr	The I^2C device address to boot from	YES
26	Boot Dev Addr Ext	Extended boot device address	YES
28	Broadcast Addr	I^2C address used to send data in the I^2C master broadcast mode.	NO
30	Local Address	The I^2C address of this device	NO
34	Bus Frequency	The desired I^2C data rate (kHz)	NO
36	Next Dev Addr	The next device address to boot (Used only if boot config option is selected)	NO
38	Next Dev Addr Ext	The extended next device address to boot (Used only if boot config option is selected)	NO
40	Address Delay	The number of CPU cycles to delay between writing the address to an I^2C EEPROM and reading data.	NO

8.1.2.4.5 SPI Boot Parameter Table

Table 8-19. SPI Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
22	Options	Bits 01 & 00 Modes • 00 = Load a boot parameter table from the SPI (Default mode) • 01 = Load boot records from the SPI (boot tables) • 10 = Load boot config records from the SPI (boot config tables) • 11 = Load GP header blob Bits 15- 02= Reserved	NO
24	Address Width	The number of bytes in the SPI device address. Can be 16 or 24 bit	YES
26	NPin	The operational mode, 4 or 5 pin	YES
28	Chipsel	The chip select used (valid in 4 pin mode only). Can be 0-3.	YES
30	Mode	Standard SPI mode (0-3)	YES
32	C2Delay	Setup time between chip assert and transaction	NO
34	Bus Freq, 100kHz	The SPI bus frequency in kHz.	NO
36	Read Addr MSW	The first address to read from, MSW (valid for 24 bit address width only)	YES
38	Read Addr LSW	The first address to read from, LSW	YES
40	Next Chip Select	Next Chip Select to be used (Used only in boot Config mode)	NO
42	Next Read Addr MSW	The Next read address (used in boot config mode only)	NO
44	Next Read Addr LSW	The Next read address (used in boot config mode only)	NO

8.1.2.4.6 HyperLink Boot Parameter Table

Table 8-20. HyperLink Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
12	Options	Bits 00 Reserved Bits 01 Configuration of Hyperlink • 0 = HyperLink is configured by RBL • 1 = HyperLink is not configured by RBL Bits 15-02 = Reserved	NO
14	Number of Lanes	Number of Lanes to be configured	NO
16	SerDes cfg msw	PCIe SerDes config word, MSW	NO
18	SerDes cfg lsw	PCIe SerDes config word, LSW	NO
20	SerDes CFG RX lane 0 cfg msw	SerDes RX lane config word, msw lane 0	NO
22	SerDes CFG RXlane 0 cfg lsw	SerDes RX lane config word, lsw, lane 0	NO
24	SerDes CFG TX lane 0 cfg msw	SerDes TX lane config word, msw lane 0	NO
26	SerDes CFG TXlane 0 cfg lsw	SerDes TX lane config word, lsw, lane 0	NO
28	SerDes CFG RX lane 1 cfg msw	SerDes RX lane config word, msw lane 1	NO
30	SerDes CFG RXlane 1 cfg lsw	SerDes RX lane config word, lsw, lane 1	NO
32	SerDes CFG TX lane 1 cfg msw	SerDes TX lane config word, msw lane 1	NO
34	SerDes CFG TXlane 1 cfg lsw	SerDes TX lane config word, lsw, lane 1	NO
36	SerDes CFG RX lane 2 cfg msw	SerDes RX lane config word, msw lane 2	NO
38	SerDes CFG RXlane 2 cfg lsw	SerDes RX lane config word, lsw, lane 2	NO
40	SerDes CFG TX lane 2 cfg msw	SerDes TX lane config word, msw lane 2	NO
42	SerDes CFG TXlane 2 cfg lsw	SerDes TX lane config word, lsw, lane 2	NO
44	SerDes CFG RX lane 3 cfg msw	SerDes RX lane config word, msw lane 3	NO
46	SerDes CFG RXlane 3 cfg lsw	SerDes RX lane config word, lsw, lane 3	NO

Table 8-20. HyperLink Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
48	SerDes CFG TX lane 3 cfg msw	SerDes TX lane config word, msw lane 3	NO
50	SerDes CFG TXlane 3 cfg lsw	SerDes TX lane config word, lsw, lane 3	NO

8.1.2.4.7 UART Boot Parameter Table

Table 8-21. UART Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
22	Reserved	None	NA
24	Data Format	Bits 00 Data Format • 0 = Data Format is BLOB • 1 = Data Format is Boot Table Bits 15 - 01 Reserved	NO
26	Protocol	Bits 00 Protocol • 0 = Xmodem Protocol • 1 = Reserved Bits 15 - 01 Reserved	NO
28	Initial NACK Count	Number of NACK pings to be sent before giving up	NO
30	Max Err Count	Maximum number of consecutive receive errors acceptable.	NO
32	NACK Timeout	Time (msecs) waiting for NACK/ACK.	NO
34	Character Timeout	Time Period between characters	NO
36	nDatabits	Number of bits supported for data. Only 8 bits is supported.	NO
38	Parity	Bits 01 - 00 Parity • 00 = No Parity • 01 = Odd parity • 10 = Even Parity Bits 15 - 02 Reserved	NO
40	nStopBitsx2	Number of stop bits times two. Valid values are 2 (stop bits = 1), 3 (Stop Bits = 1.5), 4 (Stop Bits = 2)	NO
42	Over sample factor	The over sample factor. Only 13 and 16 are valid.	NO
44	Flow Control	Bits 00 Flow Control • 0 = No Flow Control • 1 = RTS_CTS flow control Bits 15 - 01 Reserved	NO
46	Data Rate MSW	Baud Rate, MSW	NO
48	Data Rate LSW	Baud Rate, LSW	NO

8.1.2.4.8 NAND Boot Parameter Table

Table 8-22. NAND Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
22	Options	Bits 00 Geometry • 0 = Geometry is taken from this table • 1 = Geometry is queried from NAND device. Bits 01 Clear NAND • 0 = NAND Device is a non clear NAND and requires ECC • 1 = NAND is a clear NAND and doesn.t need ECC. Bits 15 - 02 Reserved	NO

Table 8-22. NAND Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
24	numColumnAddrBytes	Number of bytes used to specify column address	NO
26	numRowAddrBytes	Number of bytes used to specify row address.	NO
28	numofDataBytesperPage_msw	Number of data bytes in each page, MSW	NO
30	numofDataBytesperPage_lsw	Number of data bytes in each page, LSW	NO
32	numPagesperBlock	Number of Pages per Block	NO
34	busWidth	EMIF bus width. Only 8 or 16 bits is supported.	NO
36	numSpareBytesperPage	Number of spare bytes allocated per page.	NO
38	csel	Chip Select number (valid chip selects are 2-5)	YES
40	First Block	First block for RBL to try to read.	YES

8.1.2.4.9 DDR3 Configuration Table

The RBL also provides an option to configure the DDR table before loading the image into the external memory. More information on how to configure the DDR3, refer to the Bootloader User Guide. The configuration table for DDR3 is shown in Table 8-23

Table 8-23. DDR3 Boot Parameter Table

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
0	configselect msw	Selecting the configuration register below that to be set. Each filed below is represented by one bit each.	NO
4	configselect slsw	Selecting the configuration register below that to be set. Each filed below is represented by one bit each.	NO
8	configselect lsw	Selecting the configuration register below that to be set. Each filed below is represented by one bit each.	NO
12	pllprediv	PLL pre divider value (Should be the exact value not value -1)	NO
16	pllMult	PLL Multiplier value (Should be the exact value not value -1)	NO
20	pllPostDiv	PLL post divider value (Should be the exact value not value -1)	NO
24	sdRamConfig	SDRAM config register	NO
28	sdRamConfig2	SDRAM Config register	NO
32	sdRamRefreshctl	SDRAM Refresh Control Register	NO
36	sdRamTiming1	SDRAM Timing 1 Register	NO
40	sdRamTiming2	SDRAM Timing 2 Register	NO
44	sdRamTiming3	SDRAM Timing 3 Register	NO
48	IpDfrNvmTiming	LP DDR2 NVM Timing Register	NO
52	powerMngCtl	Power management Control Register	NO
56	iODFTTestLogic	IODFT Test Logic Global Control Register	NO
60	performcountCfg	Performance Counter Config Register	NO
64	performCountMstRegSel	Performance Counter Master Region Select Register	NO
68	readIdleCtl	Read IDLE counter Register	NO
72	sysVbusmIntEnSet	System Interrupt Enable Set Register	NO
76	sdRamOutImpdedCalcfg	SDRAM Output Impedence Calibration Config Register	NO
80	tempAlertCfg	Temperature Alert Configuration Register	NO
84	ddrPhyCtl1	DDR PHY Control Register 1	NO
88	ddrPhyCtl2	DDR PHY Control Register 1	NO
92	proClassSvceMap	Priority to Class of Service mapping Register	NO
96	mstId2ClsSvce1Map	Master ID to Class of Service Mapping 1 Register	NO
100	mstId2ClsSvce2Map	Master ID to Class of Service Mapping 2Register	NO

Table 8-23. DDR3 Boot Parameter Table (continued)

BYTE OFFSET	NAME	DESCRIPTION	CONFIGURED THROUGH BOOT CONFIGURATION PINS
104	eccCtl	ECC Control Register	NO
108	eccRange1	ECC Address Range1 Register	NO
112	eccRange2	ECC Address Range2 Register	NO
116	rdWrtExcThresh	Read Write Execution Threshold Register	NO
120 - 376	Chip Config	Chip Specific PHY configuration	NO

8.1.2.5 Second-Level Bootloaders

Any of the boot modes can be used to download a second-level bootloader. A second-level bootloader allows for:

- Any level of customization to current boot methods
- Definition of a completely customized boot

8.1.3 SoC Security

The TI SoC contains security architecture that allows the ARM CorePac to perform secure accesses within the device. For more information, contact a TI sales office for additional information available with the purchase of a secure device.

8.1.4 System PLL Settings

The PLL default settings are determined by the BOOTMODE[7:5] bits. Table 8-24 shows the settings for various input clock frequencies. This will set the PLL to the maximum clock setting for the device.

$$CLK = CLKIN \times ((PLLM+1) \div ((OUTPUT\_DIVIDE+1) \times (PLLD+1)))$$

Where OUTPUT_DIVIDE is the value of the field of SECCTL[22:19]

NOTE

Other frequencies are supported, but require a boot in a pre-configured mode.

The configuration for the NETCP PLL is also shown. The NETCP PLL is configured with these values only if the Ethernet boot mode is selected with the input clock set to match the main PLL clock (not the SGMII SerDes clock). See Table 8-9 for details on configuring Ethernet boot mode. The output from the NETCP PLL goes through an on-chip divider to reduce the frequency before reaching the NETCP. The NETCP PLL generates 1050 MHz, and after the chip divider (/3), applies 350 MHz to the NETCP.

The Main PLL is controlled using a PLL controller and a chip-level MMR. DDR3 PLL and NETCP PLL are controlled by chip level MMRs. For details on how to set up the PLL see Section 10.5. For details on the operation of the PLL controller module, see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

Table 8-24. System PLL Configuration

BOOTMODE [7:5]	INPUT CLOCK FREQ (MHz)	800 MHz DEVICE			1000 MHz DEVICE			1200 MHz DEVICE			1400 MHz DEVICE			NETCP = 350 MHz(1)		
		PLLD	PLLM	SoC *f*	PLLD	PLLM	SoC *f*	PLLD	PLLM	SoC *f*	PLLD	PLLM	SoC *f*	PLLD	PLLM	SoC *f*(2)
0b000	50.00	0	31	800	0	39	1000	0	47	1200	0	55	1400	0	41	1050
0b001	66.67	0	23	800.04	0	29	1000.05	0	35	1200.06	0	41	1400.1	1	62	1050.053
0b010	80.00	0	19	800	0	24	1000	0	29	1200	0	34	1400	3	104	1050
0b011	100.00	0	15	800	0	19	1000	0	23	1200	0	27	1400	0	20	1050
0b100	156.25	3	40	800.78	4	63	1000	2	45	1197.92	0	17	1406.3	24	335	1050
0b101	250.00	4	31	800	0	7	1000	4	47	1200	4	55	1400	4	41	1050
0b110	312.50	7	40	800.78	4	31	1000	2	22	1197.92	0	8	1406.3	24	167	1050
0b111	122.88	0	12	798.72	3	64	999.989	0	19	1228.80	0	22	1413.1	11	204	1049.6

(1) The NETCP PLL generates 1050 MHz and is internally divided by 3 to feed 350 MHz to the packet accelerator.
(2) *f* represents frequency in MHz.

8.2 Device Configuration

Certain device configurations like boot mode and endianess are selected at device power-on reset. The status of the peripherals (enabled/disabled) is determined after device power-on reset. By default, the peripherals on the device are disabled and need to be enabled by software before being used.

8.2.1 Device Configuration at Device Reset

The logic level present on each device configuration pin is latched at power-on reset to determine the device configuration. The logic level on the device configuration pins can be set by using external pullup/pulldown resistors or by using some control device (e.g., FPGA/CPLD) to intelligently drive these pins. When using a control device, care should be taken to ensure there is no contention on the lines when the device is out of reset. The device configuration pins are sampled during power-on reset and are driven after the reset is removed. To avoid contention, the control device must stop driving the device configuration pins of the SoC. Table 8-25 describes the device configuration pins.

NOTE

If a configuration pin must be routed out from the device and it is not driven (Hi-Z state), the internal pullup/pulldown (IPU/IPD) resistor should not be relied upon. TI recommends the use of an external pullup/pulldown resistor. For more detailed information on pullup/pulldown resistors and situations in which external pullup/pulldown resistors are required, see Section 5.4.

Table 8-25. Device Configuration Pins

CONFIGURATION PIN	PIN NO.	IPD/IPU(1)	DESCRIPTION
LENDIAN(1)(2)	V30	IPU	Device endian mode (LENDIAN) • 0 = Device operates in big endian mode • 1 = Device operates in little endian mode
BOOTMODE[15:0](1)(2)	AB33, AB32, AA33, AA30, Y32, Y30, AB29, W33, W31, V31, W32, W30, V32, V33, Y29, AA29	IPD	Method of boot • See Section 8.1.2 for more details.
AVSIFSEL[1:0](1)(2)	K32, K33	IPD	AVS interface selection • 00 = AVS 4-pin 6-bit Dual-Phase VCNTL[5:2] (Default) • 01 = AVS 4-pin 4-bit Single-Phase VCNTL[5:2] • 10 = AVS 6-pin 6-bit Single-Phase VCNTL[5:0] • 11 = I^2C
MAINPLLODSEL(1)(2)	Y33	IPD	Main PLL Output divider select • 0 = Main PLL output divider needs to be set to 2 by BOOTROM • 1 = Reserved
BOOTMODE_RSVD(1)	Y31	IPD	Boot Mode Reserved. Secondary function for GPIO15. Pulldown resistor required on pin.

(1) Internal 100-µA pulldown or pullup is provided for this terminal. In most systems, a 1-kΩ resistor can be used to oppose the IPD/IPU. For more detailed information on pulldown/pullup resistors and situations in which external pulldown/pullup resistors are required, see Section 5.4.
(2) These signal names are the secondary functions of these pins.

8.2.2 Peripheral Selection After Device Reset

Several of the peripherals on the AM5K2E0x are controlled by the Power Sleep Controller (PSC). By default, the PCIe and HyperLink are held in reset and clock-gated. The memories in these modules are also in a low-leakage sleep mode. Software is required to turn these memories on. Then, the software enables the modules (turns on clocks and de-asserts reset) before these modules can be used.

If one of the above modules is used in the selected ROM boot mode, the ROM code automatically enables the module.

All other modules come up enabled by default and there is no special software sequence to enable. For more detailed information on the PSC usage, see the *KeyStone Architecture Power Sleep Controller (PSC) User's Guide* (SPRUGV4).

8.2.3 Device State Control Registers

The AM5K2E0x device has a set of registers that are used to control the status of its peripherals. These registers are shown in Table 8-26.

Table 8-26. Device State Control Registers

ADDRESS START	ADDRESS END	SIZE	ACRONYM	DESCRIPTION
0x02620000	0x02620007	8B	Reserved	
0x02620008	0x02620017	16B	Reserved	
0x02620018	0x0262001B	4B	JTAGID	See Section 8.2.3.3
0x0262001C	0x0262001F	4B	Reserved	
0x02620020	0x02620023	4B	DEVSTAT	See Section 8.2.3.1
0x02620024	0x02620037	20B	Reserved	
0x02620038	0x0262003B	4B	KICK0	See Section 8.2.3.4
0x0262003C	0x0262003F	4B	KICK1	
0x02620040	0x02620043	4B	Reserved	

Table 8-26. Device State Control Registers (continued)

ADDRESS START	ADDRESS END	SIZE	ACRONYM	DESCRIPTION
0x02620044	0x02620047	4B	Reserved	
0x02620048	0x0262004B	4B	Reserved	
0x0262004C	0x0262004F	4B	Reserved	
0x02620050	0x02620053	4B	Reserved	
0x02620054	0x02620057	4B	Reserved	
0x02620058	0x0262005B	4B	Reserved	
0x0262005C	0x0262005F	4B	Reserved	
0x02620060	0x026200DF	128B	Reserved	
0x026200E0	0x0262010F	48B	Reserved	
0x02620110	0x02620117	8B	MACID	See Section 10.16
0x02620118	0x0262012F	24B	Reserved	
0x02620130	0x02620133	4B	Reserved	
0x02620134	0x02620137	4B	RESET_STAT_CLR	See Section 8.2.3.6
0x02620138	0x0262013B	4B	Reserved	
0x0262013C	0x0262013F	4B	BOOTCOMPLETE	See Section 8.2.3.7
0x02620140	0x02620143	4B	Reserved	
0x02620144	0x02620147	4B	RESET_STAT	See Section 8.2.3.5
0x02620148	0x0262014B	4B	Reserved	
0x0262014C	0x0262014F	4B	DEVCFG	See Section 8.2.3.2
0x02620150	0x02620153	4B	PWRSTATECTL	See Section 8.2.3.8
0x02620154	0x02620157	4B	Reserved	
0x02620158	0x0262015B	4B	Reserved	
0x0262015C	0x0262015F	4B	Reserved	
0x02620160	0x02620160	4B	Reserved	
0x02620164	0x02620167	4B	Reserved	
0x02620168	0x0262016B	4B	Reserved	
0x0262016C	0x0262017F	20B	Reserved	
0x02620180	0x02620183	4B	SmartReflex Class0	See Section 10.2.4
0x02620184	0x0262018F	12B	Reserved	
0x02620190	0x02620193	4B	Reserved	
0x02620194	0x02620197	4B	Reserved	
0x02620198	0x0262019B	4B	Reserved	
0x0262019C	0x0262019F	4B	Reserved	
0x026201A0	0x026201A3	4B	Reserved	
0x026201A4	0x026201A7	4B	Reserved	
0x026201A8	0x026201AB	4B	Reserved	
0x026201AC	0x026201AF	4B	Reserved	
0x026201B0	0x026201B3	4B	Reserved	
0x026201B4	0x026201B7	4B	Reserved	
0x026201B8	0x026201BB	4B	Reserved	
0x026201BC	0x026201BF	4B	Reserved	
0x026201C0	0x026201C3	4B	Reserved	
0x026201C4	0x026201C7	4B	Reserved	
0x026201C8	0x026201CB	4B	Reserved	
0x026201CC	0x026201CF	4B	Reserved	
0x026201D0	0x026201FF	48B	Reserved	
0x02620200	0x02620203	4B	Reserved	

Table 8-26. Device State Control Registers (continued)

ADDRESS START	ADDRESS END	SIZE	ACRONYM	DESCRIPTION
0x02620204	0x02620207	4B	Reserved	
0x02620208	0x0262020B	4B	Reserved	
0x0262020C	0x0262020F	4B	Reserved	
0x02620210	0x02620213	4B	Reserved	
0x02620214	0x02620217	4B	Reserved	
0x02620218	0x0262021B	4B	Reserved	
0x0262021C	0x0262021F	4B	Reserved	
0x02620220	0x0262023F	32B	Reserved	
0x02620240	0x02620243	4B	Reserved	See Section 8.2.3.11
0x02620244	0x02620247	4B	Reserved	
0x02620248	0x0262024B	4B	Reserved	
0x0262024C	0x0262024F	4B	Reserved	
0x02620250	0x02620253	4B	Reserved	
0x02620254	0x02620257	4B	Reserved	
0x02620258	0x0262025B	4B	Reserved	
0x0262025C	0x0262025F	4B	Reserved	
0x02620260	0x02620263	4B	IPCGR8	
0x02620264	0x02620267	4B	IPCGR9	
0x02620268	0x0262026B	4B	IPCGR10	
0x0262026C	0x0262026F	4B	IPCGR11	
0x02620270	0x0262027B	12B	Reserved	
0x0262027C	0x0262027F	4B	IPCGRH	See Section 8.2.3.11
0x02620280	0x02620283	4B	Reserved	See Section 8.2.3.10
0x02620284	0x02620287	4B	Reserved	
0x02620288	0x0262028B	4B	Reserved	
0x0262028C	0x0262028F	4B	Reserved	
0x02620290	0x02620293	4B	Reserved	
0x02620294	0x02620297	4B	Reserved	
0x02620298	0x0262029B	4B	Reserved	
0x0262029C	0x0262029F	4B	Reserved	
0x026202A0	0x026202A3	4B	IPCAR8	
0x026202A4	0x026202A7	4B	IPCAR9	
0x026202A8	0x026202AB	4B	IPCAR10	
0x026202AC	0x026202AF	4B	IPCAR11	
0x026202B0	0x026202BB	12B	Reserved	
0x026202BC	0x026202BF	4B	IPCARH	See Section 8.2.3.12
0x026202C0	0x026202FF	64B	Reserved	
0x02620300	0x02620303	4B	TINPSEL	See Section 8.2.3.13
0x02620304	0x02620307	4B	TOUTPSEL	See Section 8.2.3.14

Table 8-26. Device State Control Registers (continued)

ADDRESS START	ADDRESS END	SIZE	ACRONYM	DESCRIPTION
0x02620308	0x0262030B	4B	Reserved	See Section 8.2.3.15
0x0262030C	0x0262030F	4B	Reserved	
0x02620310	0x02620313	4B	Reserved	
0x02620314	0x02620317	4B	Reserved	
0x02620318	0x0262031B	4B	Reserved	
0x0262031C	0x0262031F	4B	Reserved	
0x02620320	0x02620323	4B	Reserved	
0x02620324	0x02620327	4B	Reserved	
0x02620328	0x0262032B	4B	RSTMUX8	
0x0262032C	0x0262032F	4B	RSTMUX9	
0x02620330	0x02620333	4B	RSTMUX10	
0x02620334	0x02620337	4B	RSTMUX11	
0x02620338	0x0262034F	4B	Reserved	
0x02620350	0x02620353	4B	CorePLLCTL0	See Section 10.5
0x02620354	0x02620357	4B	CorePLLCTL1	
0x02620358	0x0262035B	4B	PASSPLLCTL0	See Section 10.7
0x0262035C	0x0262035F	4B	PASSPLLCTL1	
0x02620360	0x02620363	4B	DDR3PLLCTL0	See Section 10.6
0x02620364	0x02620367	4B	DDR3PLLCTL1	
0x02620368	0x0262036B	4B	Reserved	
0x0262036C	0x0262036F	4B	Reserved	
0x02620370	0x02620373	4B	Reserved	
0x02620374	0x02620377	4B	Reserved	
0x02620378	0x0262039B	132B	Reserved	
0x0262039C	0x0262039F	4B	Reserved	
0x02620400	0x02620403	4B	ARMENDIAN_CFG0_0	See Section 8.2.3.17
0x02620404	0x02620407	4B	ARMENDIAN_CFG0_1	
0x02620408	0x0262040B	4B	ARMENDIAN_CFG0_2	
0x0262040C	0x026205FF	62B	Reserved	
0x02620600	0x026206FF	256B	Reserved	
0x02620700	0x02620703	4B	CHIP_MISC_CTL0	See Section 8.2.3.20
0x02620704	0x0262070F	12B	Reserved	
0x02620710	0x02620713	4B	SYSENDSTAT	See Section 8.2.3.22
0x02620714	0x02620717	4B	Reserved	
0x02620718	0x0262071B	4B	Reserved	
0x0262071C	0x0262071F	4B	Reserved	
0x02620720	0x0262072F	16B	Reserved	
0x02620730	0x02620733	4B	SYNECLK_PINCTL	See Section 8.2.3.23
0x02620734	0x02620737	4B	Reserved	
0x02620738	0x0262074F	24B	USB_PHY_CTL	See Section 8.2.3.24
0x02620750	0x026207FF	176B	Reserved	
0x02620800	0x02620C7B	1148B	Reserved	
0x02620C7C	0x02620C7F	4B	CHIP_MISC_CTL1	See Section 8.2.3.21
0x02620C80	0x02620C97	24B	Reserved	
0x02620C98	0x02620C9B	4B	DEVSPEED	See Section 8.2.3.16
0x02620C9C	0x02620FFF	868B	Reserved	

8.2.3.1 Device Status (DEVSTAT) Register

The Device Status Register depicts device configuration selected upon a power-on reset by the $\overline{\text{POR}}$ or RESETFULL pin. Once set, these bits remain set until a power-on reset. The Device Status Register is shown in the figure below.

Figure 8-12. Device Status Register

31 22	21 20	19	18 17	16 1	0
Reserved	Reserved	MAINPLLODSEL	AVSIFSEL	BOOTMODE	LENDIAN
R-0	R/W-00	R/W-x	R/W-xx	R/W-x xxxx xxxx xxxx xxx	R-x(1)

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

(1) x indicates the bootstrap value latched via the external pin

Table 8-27. Device Status Register Field Descriptions

Bit	Field	Description
31-22	Reserved	Reserved
21-20	Reserved	Reserved
19	MAINPLLODSEL	Main PLL Output divider select • 0 = Main PLL output divider needs to be set to 2 by BOOTROM • 1 = Reserved
18-17	AVSIFSEL	AVS interface selection • 00 = AVS 4-pin 6-bit Dual-Phase VCNTL[5:2] (Default) • 01 = AVS 4-pin 4-bit Single-Phase VCNTL[5:2] • 10 = AVS 6-pin 6-bit Single-Phase VCNTL[5:0] • 11 = Reserved
16-1	BOOTMODE	Determines the bootmode configured for the device. For more information on bootmode, see Section 8.1.2. See the *KeyStone II Architecture ARM Bootloader User's Guide* (SPRUHJ3).
0	LENDIAN	Device endian mode (LENDIAN) — shows the status of whether the system is operating in big endian mode or little endian mode (default). • 0 = System is operating in big endian mode • 1 = System is operating in little endian mode (default)

8.2.3.2 Device Configuration Register

The Device Configuration Register is one-time writeable through software. The register is reset on all hard resets and is locked after the first write. The Device Configuration Register is shown in Figure 8-13 and described in Table 8-28.

Figure 8-13. Device Configuration Register (DEVCFG)

31 5	4 3	2 1	0
Reserved	PCIE1SSMODE	PCIE0SSMODE	SYSCLKOUTEN
R-0	R/W-00	R/W-00	R/W-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 8-28. Device Configuration Register Field Descriptions

Bit	Field	Description
31-5	Reserved	Reserved. Read only, writes have no effect.
4-3	PCIE1SSMODE	Device Type Input of PCIe1SS • 00 = Endpoint • 01 = Legacy Endpoint • 10 = Rootcomplex • 11 = Reserved

Table 8-28. Device Configuration Register Field Descriptions (continued)

Bit	Field	Description
2-1	PCIE0SSMODE	Device Type Input of PCIe0SS • 00 = Endpoint • 01 = Legacy Endpoint • 10 = Rootcomplex • 11 = Reserved
0	SYSCLKOUTEN	SYSCLKOUT enable • 0 = No clock output • 1 = Clock output enabled (default)

8.2.3.3 JTAG ID (JTAGID) Register Description

The JTAG ID register is a read-only register that identifies to the customer the JTAG/Device ID. For the device, the JTAG ID register resides at address location 0x02620018. The JTAG ID Register is shown below.

Figure 8-14. JTAG ID (JTAGID) Register

31 28	27 12	11 1	0
VARIANT	PART NUMBER	MANUFACTURER	LSB
R-xxxx	R-1011 1001 1010 0110	R-0000 0010 111	R-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 8-29. JTAG ID Register Field Descriptions

Bit	Field	Value	Description
31-28	VARIANT	xxxx	Variant value
27-12	PART NUMBER	1011 1001 1010 0110	Part Number for boundary scan
11-1	MANUFACTURER	0000 0010 111	Manufacturer
0	LSB	1	This bit is read as a 1

NOTE

The value of the VARIANT and PART NUMBER fields depends on the silicon revision being used. See the Silicon Errata for details.

8.2.3.4 Kicker Mechanism (KICK0 and KICK1) Register

The Bootcfg module contains a kicker mechanism to prevent spurious writes from changing any of the Bootcfg MMR (memory mapped registers) values. When the kicker is locked (which it is initially after power on reset), none of the Bootcfg MMRs are writable (they are only readable). This mechanism requires an MMR write to each of the KICK0 and KICK1 registers with exact data values before the kicker lock mechanism is unlocked. See Table 8-26 for the address location. Once released, all the Bootcfg MMRs having write permissions are writable (the read only MMRs are still read only). The KICK0 data is 0x83e70b13. The KICK1 data is 0x95a4f1e0. Writing any other data value to either of these kick MMRs locks the kicker mechanism and blocks writes to Bootcfg MMRs. To ensure protection to all Bootcfg MMRs, software must always re-lock the kicker mechanism after completing the MMR writes.

8.2.3.5 Reset Status (RESET_STAT) Register

The Reset Status Register (RESET_STAT) captures the status of global device reset (GR). Software can use this information to take different device initialization steps. The GR bit is written as 1 only when a global reset is asserted.

The Reset Status Register is shown in the figure and table below.

Figure 8-15. Reset Status Register (RESET_STAT)

31	30	0
GR	Reserved	
R-1	R- 0	

Legend: R = Read only; -*n* = value after reset

Table 8-30. Reset Status Register Field Descriptions

Bit	Field	Description
31	GR	Global reset status • 0 = Device has not received a global reset. • 1 = Device received a global reset.
30-0	Reserved	Reserved.

8.2.3.6 Reset Status Clear (RESET_STAT_CLR) Register

The RESET_STAT bits can be cleared by writing 1 to the corresponding bit in the RESET_STAT_CLR register. The Reset Status Clear Register is shown in the figure and table below.

Figure 8-16. Reset Status Clear Register (RESET_STAT_CLR)

31	30	1	0
GR	Reserved		
RW-0	R- 0		

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-31. Reset Status Clear Register Field Descriptions

Bit	Field	Description
31	GR	Global reset clear bit • 0 = Writing a 0 has no effect. • 1 = Writing a 1 to the GR bit clears the corresponding bit in the RESET_STAT register.
30-0	Reserved	Reserved.

8.2.3.7 Boot Complete (BOOTCOMPLETE) Register

The BOOTCOMPLETE register controls the BOOTCOMPLETE pin status to indicate the completion of the ROM booting process. The Boot Complete register is shown in the figure and table below.

Figure 8-17. Boot Complete Register (BOOTCOMPLETE)

31	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		BC11	BC10	BC9	BC8	Reserved							
R-0		RW-0	RW-0	RW-0	RW-0	R-0							

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-32. Boot Complete Register Field Descriptions

Bit	Field	Description
31-12	Reserved	
11	BC11	ARM CorePac 3 boot status (AM5K2E04 only) • 0 = ARM CorePac 3 boot NOT complete • 1 = ARM CorePac 3 boot complete
10	BC10	ARM CorePac 2 boot status (AM5K2E04 only) • 0 = ARM CorePac 2 boot NOT complete • 1 = ARM CorePac 2 boot complete
9	BC9	ARM CorePac 1 boot status (AM5K2Ex) • 0 = ARM CorePac 1 boot NOT complete • 1 = ARM CorePac 1 boot complete
8	BC8	ARM CorePac 0 boot status • 0 = ARM CorePac 0 boot NOT complete • 1 = ARM CorePac 0 boot complete
7-0	Reserved	

The BCx bit indicates the boot complete status of the corresponding ARM CorePac. All BCx bits are sticky bits — that is, they can be set only once by the software after device reset and they will be cleared to 0 on all device resets (warm reset and power-on reset).

Boot ROM code is implemented such that each ARM CorePac sets its corresponding BCx bit immediately before branching to the predefined location in memory.

8.2.3.8 Power State Control (PWRSTATECTL) Register

The Power State Control Register (PWRSTATECTL) is controlled by the software to indicate the power-saving mode. Under ROM code, the CorePac reads this register to differentiate between the various power saving modes. This register is cleared only by POR and is not changed by any other device reset. See the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0) for more information. The PWRSTATECTL register is shown in Figure 8-18 and described in Table 8-33.

Figure 8-18. Power State Control Register (PWRSTATECTL)

31	3	2	1	0
Hibernation Recovery Branch Address		Hibernation Mode	Hibernation	Standby
RW-0000 0000 0000 0000 0		RW-0	RW-0	RW-0

Legend: R = Read Only, RW = Read/Write; -*n* = value after reset

Table 8-33. Power State Control Register Field Descriptions

Bit	Field	Description
31-3	Hibernation Recovery Branch Address	Used to provide a start address for execution out of the hibernation modes.
2	Hibernation Mode	Indicates whether the device is in hibernation mode 1 or mode 2. • 0 = Hibernation mode 1 • 1 = Hibernation mode 2
1	Hibernation	Indicates whether the device is in hibernation mode or not. • 0 = Not in hibernation mode • 1 = Hibernation mode
0	Standby	Indicates whether the device is in standby mode or not. • 0 = Not in standby mode • 1 = standby mode

8.2.3.9 IPC Generation (IPCGRx) Registers

The IPCGRx Registers facilitate inter-C66x CorePac interrupts.

The AM5K2E device has four IPCGRx registers (IPCGR8-IPCGR11) and the 66AK2E02 has two IPCGRx registers (IPCGR8 and IPCGR9). These registers can be used by external hosts or CorePacs to generate interrupts to other CorePacs. A write of 1 to the IPCG field of the IPCGRx register generates an interrupt pulse to the ARM CorePac.

These registers also provide a *Source ID* facility identifying up to 28 different sources of interrupts. Allocation of source bits to source processor and meaning is entirely based on software convention. The register field descriptions are given in the following tables. There can be numerous sources for these registers as this is completely controlled by software. Any master that has access to BOOTCFG module space can write to these registers. The IPC Generation Register is shown in Figure 8-19 and described in Table 8-34.

Figure 8-19. IPC Generation Registers (IPCGRx)

31	4	3	1	0
SRCS27 - SRCS0		Reserved		IPCG
RW +0 (per bit field)		R-000		RW-0

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-34. IPC Generation Registers Field Descriptions

Bit	Field	Description
31-4	SRCSx	Reads return current value of internal register bit. Writes: • 0 = No effect • 1 = Sets both SRCSx and the corresponding SRCCx.
3-1	Reserved	Reserved
0	IPCG	Reads return 0. Writes: • 0 = No effect • 1 = Creates an inter-ARM interrupt.

8.2.3.10 IPC Acknowledgment (IPCARx) Registers

The IPCARx registers facilitate inter-CorePac interrupt acknowledgment.

The AM5K2E04 device has four IPCARx registers and the AM5K02 has two IPCARx registers. These registers also provide a *Source ID* facility by which up to 28 different sources of interrupts can be identified. Allocation of source bits to source processor and meaning is entirely based on software convention. The register field descriptions are given in the following tables. Virtually anything can be a source for these registers as this is completely controlled by software. Any master that has access to BOOTCFG module space can write to these registers. The IPC Acknowledgment Register is shown in the following figure and table.

Figure 8-20. IPC Acknowledgment Registers (IPCARx)

31	4	3	0
SRCC27 - SRCC0		Reserved	
RW +0 (per bit field)		R-0000	

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-35. IPC Acknowledgment Registers Field Descriptions

Bit	Field	Description
31-4	SRCCx	Reads return current value of internal register bit. Writes: • 0 = No effect • 1 = Clears both SRCCx and the corresponding SRCSx
3-0	Reserved	Reserved

8.2.3.11 IPC Generation Host (IPCGRH) Register

The IPCGRH register facilitates interrupts to external hosts. Operation and use of the IPCGRH register is the same as for other IPCGR registers. The interrupt output pulse created by the IPCGRH register appears on device pin HOUT.

The host interrupt output pulse is stretched so that it is asserted for four bootcfg clock cycles (SYSCLK1/6) followed by a deassertion of four bootcfg clock cycles. Generating the pulse results in a pulse-blocking window that is eight SYSCLK1/6-cycles long. Back-to-back writes to the IPCRGH register with the IPCG bit (bit 0) set, generates only one pulse if the back-to-back writes to IPCGRH are less than the eight SYSCLK1/6 cycle window — the pulse blocking window. To generate back-to-back pulses, the back-to-back writes to the IPCGRH register must be written after the eight SYSCLK1/6 cycle pulse-blocking window has elapsed. The IPC Generation Host Register is shown in Figure 8-21 and described in Table 8-36.

Figure 8-21. IPC Generation Registers (IPCGRH)

31	4	3	1	0
SRCS27 - SRCS0		Reserved		IPCG
RW +0 (per bit field)		R-000		RW +0

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-36. IPC Generation Registers Field Descriptions

Bit	Field	Description
31-4	SRCSx	Reads return current value of internal register bit. Writes: • 0 = No effect • 1 = Sets both SRCSx and the corresponding SRCCx.
3-1	Reserved	Reserved
0	IPCG	Reads return 0. Writes: • 0 = No effect • 1 = Creates an interrupt pulse on device pin (host interrupt/event output in HOUT pin)

8.2.3.12 IPC Acknowledgment Host (IPCARH) Register

The IPCARH register facilitates external host interrupts. Operation and use of the IPCARH register is the same as for other IPCAR registers. The IPC Acknowledgment Host Register is shown in Figure 8-22 and described in Table 8-37.

Figure 8-22. Acknowledgment Register (IPCARH)

31	4	3	0
SRCC27 - SRCC0		Reserved	
RW +0 (per bit field)		R-0000	

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-37. IPC Acknowledgment Register Field Descriptions

Bit	Field	Description
31-4	SRCCx	Reads the return current value of the internal register bit. Writes: • 0 = No effect • 1 = Clears both SRCCx and the corresponding SRCSx
3-0	Reserved	Reserved

8.2.3.13 Timer Input Selection Register (TINPSEL)

The Timer Input Selection Register selects timer inputs and is shown in Figure 8-23 and described in Table 8-38.

Figure 8-23. Timer Input Selection Register (TINPSEL)

31	30	29	28	27	26	25	24
TINPHSEL15	TINPLSEL15	TINPHSEL14	TINPLSEL14	TINPHSEL13	TINPLSEL13	TINPHSEL12	TINPLSEL12
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
23	22	21	20	19	18	17	16
TINPHSEL11	TINPLSEL11	TINPHSEL10	TINPLSEL10	TINPHSEL9	TINPLSEL9	TINPHSEL8	TINPLSEL8
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

15	0
Reserved	
R-0	

LEGEND: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-38. Timer Input Selection Field Description

Bit	Field	Description
31	TINPHSEL15	Input select for TIMER15 high. • 0 = TIMI0 • 1 = TIMI1
30	TINPLSEL15	Input select for TIMER15 low. • 0 = TIMI0 • 1 = TIMI1
29	TINPHSEL14	Input select for TIMER14 high. • 0 = TIMI0 • 1 = TIMI1
28	TINPLSE14	Input select for TIMER14 low. • 0 = TIMI0 • 1 = TIMI1
27	TINPHSEL13	Input select for TIMER13 high. • 0 = TIMI0 • 1 = TIMI1
26	TINPLSEL13	Input select for TIMER13 low. • 0 = TIMI0 • 1 = TIMI1
25	TINPHSEL12	Input select for TIMER12 high. • 0 = TIMI0 • 1 = TIMI1
24	TINPLSEL12	Input select for TIMER12low. • 0 = TIMI0 • 1 = TIMI1
23	TINPHSEL11	Input select for TIMER11 high. • 0 = TIMI0 • 1 = TIMI1
22	TINPLSEL11	Input select for TIMER11 low. • 0 = TIMI0 • 1 = TIMI1
21	TINPHSEL10	Input select for TIMER10 high. • 0 = TIMI0 • 1 = TIMI1
20	TINPLSEL10	Input select for TIMER10 low. • 0 = TIMI0 • 1 = TIMI1

Table 8-38. Timer Input Selection Field Description (continued)

Bit	Field	Description
19	TINPHSEL9	Input select for TIMER9 high. • 0 = TIMI0 • 1 = TIMI1
18	TINPLSEL9	Input select for TIMER9 low. • 0 = TIMI0 • 1 = TIMI1
17	TINPHSEL8	Input select for TIMER8 high. • 0 = TIMI0 • 1 = TIMI1
16	TINPLSEL8	Input select for TIMER8 low. • 0 = TIMI0 • 1 = TIMI1
15-0	Reserved	

8.2.3.14 Timer Output Selection Register (TOUTPSEL)

The control register TOUTSEL handles the timer output selection and is shown in Figure 8-24 and described in Table 8-39.

Figure 8-24. Timer Output Selection Register (TOUTPSEL)

31	10	9	5	4	0
Reserved		TOUTPSEL1		TOUTPSEL0	
R-00000000000000000000		RW-00001		RW-00000	

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-39. Timer Output Selection Field Description

Bit	Field	Description
31-10	Reserved	Reserved
9-5	TOUTPSEL1	Output select for TIMO1 • 00000: Reserved • 00001: Reserved • 00010: Reserved • 00011: Reserved • 00100: Reserved • 00101: Reserved • 00110: Reserved • 00111: Reserved • 01000: Reserved • 01001: Reserved • 01010: Reserved • 01011: Reserved • 01100: Reserved • 01101: Reserved • 01110: Reserved • 01111: Reserved • 10000: TOUTL8 • 10001: TOUTH8 • 10010: TOUTL9 • 10011: TOUTH9 • 10100: TOUTL10 • 10101: TOUTH10 • 10110: TOUTL11 • 10111: TOUTH11 • 11000: TOUTL12 • 11001: TOUTH12 • 11010: TOUTL13 • 11011: TOUTH13 • 11100: TOUTL14 • 11101: TOUTH14 • 11110: TOUTL15 • 11111: TOUTH15

Table 8-39. Timer Output Selection Field Description (continued)

Bit	Field	Description
4-0	TOUTPSEL0	Output select for TIMO0 • 00000: Reserved • 00001: Reserved • 00010: Reserved • 00011: Reserved • 00100: Reserved • 00101: Reserved • 00110: Reserved • 00111: Reserved • 01000: Reserved • 01001: Reserved • 01010: Reserved • 01011: Reserved • 01100: Reserved • 01101: Reserved • 01110: Reserved • 01111: Reserved • 10000: TOUTL8 • 10001: TOUTH8 • 10010: TOUTL9 • 10011: TOUTH9 • 10100: TOUTL10 • 10101: TOUTH10 • 10110: TOUTL11 • 10111: TOUTH11 • 11000: TOUTL12 • 11001: TOUTH12 • 11010: TOUTL13 • 11011: TOUTH13 • 11100: TOUTL14 • 11101: TOUTH14 • 11110: TOUTL15 • 11111: TOUTH15

8.2.3.15 Reset Mux (RSTMUXx) Register

Software controls the Reset Mux block through the reset multiplex registers using RSTMUX8-RSTMUX11 for the ARM CorePac (AM5K2E04) or RSTMUX8_RSTMUX9 for the ARM CorePac (AM5K2E02) on the device. These registers are located in Bootcfg memory space. The Reset Mux Register is shown in Figure 8-25 and Table 8-40 below.

Figure 8-25. Reset Mux Register

31 10	9	8	7 5	4	3 1	0
Reserved	EVTSTATCLR	Reserved	DELAY	EVTSTAT	OMODE	LOCK
R-0000 0000 0000 0000 0000 00	RC-0	R-0	RW-100	R-0	RW-000	RW-0

Legend: R = Read only; RW = Read/Write; -*n* = value after reset; RC = Read only and write 1 to clear

Table 8-40. Reset Mux Register 8..11(RSTMUX8-RSTMUX11) Field Descriptions

Bit	Field	Description
31-10	Reserved	Reserved
9	EVTSTATCLR	Clear event status • 0 = Writing 0 has no effect • 1 = Writing 1 to this bit clears the EVTSTAT bit
8	Reserved	Reserved
7-5	DELAY	Delay cycles between interrupt and device reset • 000b = 256 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b • 001b = 512 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b • 010b = 1024 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b • 011b = 2048 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b • 100b = 4096 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b (default) • 101b = 8192 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b • 110b = 16384 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b • 111b = 32768 SYSCLK1/6 cycles delay between interrupt and device reset, when OMODE = 100b
4	EVTSTAT	Event status • 0 = No event received (Default) • 1 = WD timer event received by Reset Mux block

Table 8-40. Reset Mux Register 8..11(RSTMUX8-RSTMUX11) Field Descriptions (continued)

Bit	Field	Description
3-1	OMODE	Timer event operation mode • 000b = WD timer event input to the Reset Mux block does not cause any output event (default) • 001b = Reserved • 010b = Cortex-A15 processor watchdog timers, the Local Reset output event of the RSTMUX logic generates reset to PLL Controller. • 011b = WD Timer Event input to the Reset Mux block causes Local Reset output event of the RSTMUX logic to generate reset to PLL Controller. • 100b = WD Timer Event input to the Reset Mux block causes an interrupt to be sent to the GIC. • 101b = WD timer event input to the Reset Mux block causes device reset to AM5K2E0x. Note that for Cortex-A15 processor watchdog timers, the Local Reset output event of the RSTMUX logic is connected to the Device Reset generation to generate reset to PLL Controller. • 110b = Reserved • 111b = Reserved
0	LOCK	Lock register fields • 0 = Register fields are not locked (default) • 1 = Register fields are locked until the next timer reset

8.2.3.16 Device Speed (DEVSPEED) Register

The Device Speed Register shows the device speed grade and is shown below.

Figure 8-26. Device Speed Register (DEVSPEED)

31 28	27 16	15 12	11 0
Reserved	DEVSPEED	Reserved	ARMSPEED
	R-n		R-n

Legend: R = Read only; -*n* = value after reset

Table 8-41. Device Speed Register Field Descriptions

Bit	Field	Description
31-28	Reserved	Reserved. Read only
27-16	DEVSPEED	Indicates the speed of the device (read only) • 0b0000 0000 0000 = 800 MHz • 0b0000 0000 0001 = 1000 MHz • 0b0000 0000 001x = 1200 MHz • 0b0000 0000 01xx = 1350 MHz • 0b0000 0000 1xxx = 1400 MHz • 0b0000 0001 xxxx = 1500 MHz • 0b0000 001x xxxx = 1400 MHz • 0b0000 01xx xxxx = 1350.8 MHz • 0b0000 1xxx xxxx = 1200 MHz • 0b0001 xxxx xxxx= 1000 MHz • 0b001x xxxx xxxx = 800 MHz
15-12	Reserved	Reserved. Read only

Table 8-41. Device Speed Register Field Descriptions (continued)

Bit	Field	Description
11-0	ARMSPEED	Indicates the speed of the ARM (read only) • 0b0000 0000 0000 = 800 MHz • 0b0000 0000 0001 = 1000 MHz • 0b0000 0000 001x = 1200 MHz • 0b0000 0000 01xx = 1350 MHz • 0b0000 0000 1xxx = 1400 MHz • 0b0000 0001 xxxx = 1500 MHz • 0b0000 001x xxxx = 1400 MHz • 0b0000 01xx xxxx = 1350.8 MHz • 0b0000 1xxx xxxx = 1200 MHz • 0b0001 xxxx xxxx= 1000 MHz • 0b001x xxxx xxxx = 800 MHz

8.2.3.17 ARM Endian Configuration Register 0 (ARMENDIAN_CFGr_0), r=0..7

The registers defined in ARM Configuration Register 0 (ARMENDIAN_CFGr_0) and ARM Configuration Register 1 (ARMENDIAN_CFGr_1) control the way Cortex-A15 processor core access to peripheral MMRs shows up in the Cortex-A15 processor registers. The purpose is to provide an endian-invariant view of the peripheral MMRs when performing a 32-bit access. (Only one of the eight register sets is shown.)

Figure 8-27. ARM Endian Configuration Register 0 (ARMENDIAN_CFGr_0), r=0..7

31	8	7	0
BASEADDR		Reserved	
RW		R-0000 0000	

Legend: RW = Read/Write; R = Read only

Table 8-42. ARM Endian Configuration Register 0 Default Values

ARM ENDIAN CONFIGURATION REGISTER 0	DEFAULT VALUES
ARMENDIAN_CFG0_0	0x0001C000
ARMENDIAN_CFG1_0	0x00020000
ARMENDIAN_CFG2_0	0x000BC000
ARMENDIAN_CFG3_0	0x00210000
ARMENDIAN_CFG4_0	0x00023A00
ARMENDIAN_CFG5_0	0x00240000
ARMENDIAN_CFG6_0	0x01000000
ARMENDIAN_CFG7_0	0xFFFFFF00

Table 8-43. ARM Endian Configuration Register 0 Field Descriptions

Bit	Field	Description
31-8	BASEADDR	24-bit Base Address of Configuration Region R This base address defines the start of a contiguous block of Memory Mapped Register space for which a word swap is done by the ARM CorePac bridge.
7-0	Reserved	Reserved

8.2.3.18 ARM Endian Configuration Register 1 (ARMENDIAN_CFGr_1), r=0..7

Figure 8-28. ARM Endian Configuration Register 1 (ARMENDIAN_CFGr_1), r=0..7

31	4	3	0
Reserved		SIZE	
R-0000 0000 0000 0000 0000 0000 0000		RW	

Legend: RW = Read/Write; R = Read only

Table 8-44. ARM Endian Configuration Register 1 Default Values

ARM ENDIAN CONFIGURATION REGISTER 1	DEFAULT VALUES
ARMENDIAN_CFG0_1	0x00000006
ARMENDIAN_CFG1_1	0x00000009
ARMENDIAN_CFG2_1	0x00000004
ARMENDIAN_CFG3_1	0x00000008
ARMENDIAN_CFG4_1	0x00000005
ARMENDIAN_CFG5_1	0x00000006
ARMENDIAN_CFG6_1	0x00000000
ARMENDIAN_CFG7_1	0x00000000

Table 8-45. ARM Endian Configuration Register 1 Field Descriptions

Bit	Field	Description
31-4	Reserved	Reserved
3-0	SIZE	4-bit encoded size of Configuration Region R The value in the SIZE field defines the size of the contiguous block of Memory Mapped Register space for which a word swap is done by the ARM CorePac bridge (starting from ARMENDIAN_CFGr_0.BASEADDR). • 0000 : 64KB • 0001 : 128KB • 0010 : 256KB • 0011 : 512KB • 0100 : 1MB • 0101 : 2MB • 0110 : 4MB • 0111 : 8MB • 1000 : 16MB • 1001 : 32MB • 1010 : 64MB • 1011 : 128MB • Others : Reserved

8.2.3.19 ARM Endian Configuration Register 2 (ARMENDIAN_CFGr_2), r=0..7

The registers defined in ARM Configuration Register 2 (ARMENDIAN_CFGr_2) enable the word swapping of a region.

Figure 8-29. ARM Endian Configuration Register 2 (ARMENDIAN_CFGr_2), r=0..7

31	1	0
Reserved		DIS
R-0000 0000 0000 0000 0000 0000 0000 000		RW-0

Legend: RW = Read/Write

Table 8-46. ARM Endian Configuration Register 2 Default Values

ARM ENDIAN CONFIGURATION REGISTER 2	DEFAULT VALUES
ARMENDIAN_CFG0_2	0x00000001
ARMENDIAN_CFG1_2	0x00000001
ARMENDIAN_CFG2_2	0x00000001
ARMENDIAN_CFG3_2	0x00000001
ARMENDIAN_CFG4_2	0x00000001
ARMENDIAN_CFG5_2	0x00000001
ARMENDIAN_CFG6_2	0x00000001
ARMENDIAN_CFG7_2	0x00000001

Table 8-47. ARM Endian Configuration Register 2 Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved
0	DIS	Disabling the word swap of a region • 0 : Enable word swap for region • 1 : Disable word swap for region

8.2.3.20 Chip Miscellaneous Control (CHIP_MISC_CTL0) Register

Figure 8-30. Chip Miscellaneous Control Register (CHIP_MISC_CTL0)

31	19	18
Reserved		USB_PME_EN
R-0		RW-0

17	13	12	11	3	2	0
Reserved		MSMC_BLOCK_PARITY_RST	Reserved		QM_PRIORITY	
RW -0		RW-0	RW-0		RW-0	

Legend: R = Read only; W = Write only; -*n* = value after reset

Table 8-48. Chip Miscellaneous Control Register (CHIP_MISC_CTL0) Field Descriptions

Bit	Field	Description
31-19	Reserved	Reserved.
18	USB_PME_EN	Enables wakeup event generation from USB • 0 = Disable PME event generation • 1 = Enable PME event generation
17-13	Reserved	
12	MSMC_BLOCK_PARITY_RST	Controls MSMC parity RAM reset. When set to ‘1’ means the MSMC parity RAM will not be reset.
11-3	Reserved	Reserved
2-0	QM_PRIORITY	Control the priority level for the transactions from QM Master port, which access the external linking RAM.

8.2.3.21 Chip Miscellaneous Control (CHIP_MISC_CTL1) Register

Figure 8-31. Chip Miscellaneous Control Register (CHIP_MISC_CTL1)

31	15	14	13	0
Reserved		IO_TRACE_SEL	Reserved	
R- 0000 0000 00000000		RW-0	RW-0	

Legend: R = Read only; RW = Read/Write; -*n* = value after reset

Table 8-49. Chip Miscellaneous Control Register (CHIP_MISC_CTL1) Field Descriptions

Bit	Field	Description
31-15	Reserved	Reserved.
14	IO_TRACE_SEL	This bit controls the pin muxing of GPIO[31:17] and EMU[33:19] pin • 0 = GPIO[31:17] is selected • 1 = EMU[33:19] pins is selected
13-0	Reserved	

8.2.3.22 System Endian Status Register (SYSENDSTAT)

This register provides a way for reading the system endianness in an endian-neutral way. A zero value indicates big endian and a non-zero value indicates little endian. The SYSENDSTAT register captures the LENDIAN bootmode pin and is used by the BOOTROM to guide the bootflow. The value is latched on the rising edge of POR or RESETFULL .

Figure 8-32. System Endian Status Register

31	1	0
Reserved		SYSENDSTAT
R-0000 0000 0000 0000 0000 0000 0000 000		R-0

Legend: RW = Read/Write; -*n* = value after reset

Table 8-50. System Endian Status Register Descriptions

Bit	Field	Description
31-1	Reserved	Reserved
0	SYSENDSTAT	Reflects the same value as the LENDIAN bit in the DEVSTAT register. • 0 - SoC is in Big Endian • 1 - SoC is in Little Endian

8.2.3.23 SYNECLK_PINCTL Register

This register controls the routing of recovered clock signals from any Ethernet port (SGMII/XFI of the multiport switches) to the clock output TSRXCLKOUT0/TSRXCLKOUT1.

Figure 8-33. SYNECLK_PINCTL Register

31	7	6	4	3	2	0
Reserved		TSRXCLKOUT1SEL		Reserved	TSRXCLKOUT0SEL	
R-0000 0000 0000 0000 0000 0000 0		RW-0			RW-0	

Legend: RW = Read/Write; - *n* = value after reset

Table 8-51. SYNECLK_PINCTL Register Descriptions

Bit	Field	Description
31-7	Reserved	Reserved
6-4	TSRXCLKOUT1SEL	• 000 - SGMII Lane 0 rxbclk • 001 - SGMII Lane 1 rxbclk • 010 - SGMII Lane 2 rxbclk • 011 - SGMII Lane 3 rxbclk • 100 - XFI Lane 0 rxbclk • 101 - XFI Lane 1 rxbclk • 110 - XFI Lane 2 rxbclk • 111 - XFI Lane 3 rxbclk
3	Reserved	Reserved

Table 8-51. SYNECLK_PINCTL Register Descriptions (continued)

Bit	Field	Description
2-0	TSRXCLKOUT0SEL	• 000 - SGMII Lane 0 rxbclk • 001 - SGMII Lane 1 rxbclk • 010 - SGMII Lane 2 rxbclk • 011 - SGMII Lane 3 rxbclk • 100 - XFI Lane 0 rxbclk • 101 - XFI Lane 1 rxbclk • 110 - XFI Lane 2 rxbclk • 111 - XFI Lane 3 rxbclk

8.2.3.24 USB PHY Control (USB_PHY_CTLx) Registers

The following registers control the USB PHY.

Figure 8-34. USB_PHY_CTL0 Register

31	12	11
Reserved		PHY_RTUNE_ACK
R-0		R-0

10	9	8	7	6	5
PHY_RTUNE_REQ	Reserved	PHY_TC_VATESTENB		PHY_TC_TEST_POWERDOWN_SSP	PHY_TC_TEST_POWERDOWN_HSP
R/W-0	R-0	R/W-00		R/W-0	R/W-0

4	3	2	1	0
PHY_TC_LOOPBACKENB	Reserved	UTMI_VBAUSVLDEXT	UTMI_TXBITSTUFFENH	UTMI_TXBITSTUFFEN
R/W-0	R-0	R/W-0	R/W-0	R/W-0

Legend: R = Read only; W = Write only; -*n* = value after reset

Table 8-52. USB_PHY_CTL0 Register Field Descriptions

Bit	Field	Description
31-12	Reserved	Reserved
11	PHY_RTUNE_ACK	The PHY uses an external resistor to calibrate the termination impedances of the PHY's high-speed inputs and outputs. The resistor is shared between the USB2.0 high-speed outputs and the Super-speed I/O. Each time the PHY is taken out of a reset, a termination calibration is performed. For SS link, the calibration can also be requested externally by asserting the PHY_RTUNE_REQ. When the calibration is complete, the PHY_RTUNE_ACK transitions low. A resistor calibration on the SS link cannot be performed while the link is operational
10	PHY_RTUNE_REQ	See PHY_RTUNE_ACK.
9	Reserved	Reserved
8-7	PHY_TC_VATESTENB	Analog Test Pin Select. Enables analog test voltages to be placed on the ID pin. • 11 = Invalid setting. • 10 = Invalid setting. • 01 = Analog test voltages can be viewed or applied on ID. • 00 = Analog test voltages cannot be viewed or applied on ID.
6	PHY_TC_TEST_POWERDOWN_SSP	SS Function Circuits Power-Down Control. Powers down all SS function circuitry in the PHY for IDDQ testing.
5	PHY_TC_TEST_POWERDOWN_HSP	HS Function Circuits Power-Down Control Powers down all HS function circuitry in the PHY for IDDQ testing.

Table 8-52. USB_PHY_CTL0 Register Field Descriptions (continued)

Bit	Field	Description
4	PHY_TC_LOOPBACKENB	Loop-back Test Enable Places the USB3.0 PHY in HS Loop-back mode, which concurrently enables the HS receive and transmit logic. • 1 = During HS data transmission, the HS receive logic is enabled. • 0 = During HS data transmission, the HS receive logic is disabled.
3	Reserved	• Reserved
2	UTMI_VBAUSVLDEXT	External VBUS Valid Indicator Function: Valid in Device mode and only when the VBUSVLDEXTSEL signal is set to 1'b1. VBUSVLDEXT indicates whether the VBUS signal on the USB cable is valid. In addition, VBUSVLDEXT enables the pull-up resistor on the D+ line. • 1 = VBUS signal is valid, and the pull-up resistor on D+ is enabled. • 0 = VBUS signal is not valid, and the pull-up resistor on D+ is disabled.
1	UTMI_TXBITSTUFFENH	High-byte Transmit Bit-Stuffing Enable Function: controls bit stuffing on DATAINH[7:0] when OPMODE[1:0]=11b. • 1 = Bit stuffing is enabled. • 0 = Bit stuffing is disabled.
0	UTMI_TXBITSTUFFEN	Low-byte Transmit Bit-Stuffing Enable Function: controls bit stuffing on DATAIN[7:0] when OPMODE[1:0]=11b. • 1 = Bit stuffing is enabled. • 0 = Bit stuffing is disabled.

Figure 8-35. USB_PHY_CTL1 Register

31 … 6	5
Reserved	PIPE_REF_CLKREQ_N
R-0	R-0

4	3	2	1	0
PIPE_TX2RX_LOOPBK	PIPE_EXT_PCLK_REQ	PIPE_ALT_CLK_SEL	PIPE_ALT_CLK_REQ	PIPE_ALT_CLK_EN
R/W-0	R/W-0	R/W-0	R-0	R/W-0

Legend: R = Read only; R/W = Read/Write, -*n* = value after reset

Table 8-53. USB_PHY_CTL1 Register Field Descriptions

Bit	Field	Description
31-6	Reserved	Reserved
5	PIPE_REF_CLKREQ_N	Reference Clock Removal Acknowledge. When the pipeP_power-down control into the PHY turns off the MPLL in the P3 state, PIPE_REF_CLKREQ_N is asserted after the PLL is stable and the reference clock can be removed.
4	PIPE_TX2RX_LOOPBK	Loop-back. When this signal is asserted, data from the transmit predriver is looped back to the receiver slicers. LOS is bypassed and based on the tx_en input so that rx_los=!tx_data_en.
3	PIPE_EXT_PCLK_REQ	External PIPE Clock Enable Request. When asserted, this signal enables the pipeP_pclk output regardless of power state (along with the associated increase in power consumption).
2	PIPE_ALT_CLK_SEL	Alternate Clock Source Select. Selects the alternate clock sources instead of the internal MPLL outputs for the PCS clocks. • 1 = Uses alternate clocks. • 0 = Users internal MPLL clocks. Change only during a reset.

Table 8-53. USB_PHY_CTL1 Register Field Descriptions (continued)

Bit	Field	Description
1	PIPE_ALT_CLK_REQ	Alternate Clock Source Request. Indicates that the alternate clocks are needed by the slave PCS (that is, to boot the master MPLL). Connect to the alt_clk_en on the master.
0	PIPE_ALT_CLK_EN	Alternate Clock Enable. Enables the ref_pcs_clk and ref_pipe_pclk output clocks (if necessary, powers up the MPLL).

Figure 8-36. USB_PHY_CTL2 Register

31 30	29 27	26 23	22 21
Reserved	PHY_PC_LOS_BIAS	PHY_PC_TXVREFTUNE	PHY_PC_TXRISETUNE
R-0	R/W-101	R/W-1000	R/W-01

20	19 18	17 16	15 14
PHY_PC_TXRESTUNE	PHY_PC_ TXPREEMPPULSETUNE	PHY_PC_TXPREEMPAMPTUNE	PHY_PC_ TXHSXVTUNE
R/W-01	R/W-0	R/W-00	R/W-11

13 10	9 7	6 4	3	2 0
PHY_PC_TXFSLSTUNE	PHY_PC_SQRXTUNE	PHY_PC_OTGTUNE	Reserved	PHY_PC_ COMPDISTUNE
R/W-0011	R/W-011	R/W-100	R-0	R/W-100

Legend: R = Read only; R/W = Read/Write, -*n* = value after reset

Table 8-54. USB_PHY_CTL2 Register Field Descriptions

Bit	Field	Description
31-30	Reserved	Reserved
29-27	PHY_PC_LOS_BIAS	Loss-of-Signal Detector Threshold Level Control. Sets the LOS detection threshold level. • +1 = results in a +15 mVp incremental change in the LOS threshold. • -1 = results in a -15 mVp incremental change in the LOS threshold. Note: the 000b setting is reserved and must not be used.
26-23	PHY_PC_TXVREFTUNE	HS DC Voltage Level Adjustment. Adjusts the high-speed DC level voltage. • +1 = results in a +1.25% incremental change in high-speed DC voltage level. • -1 = results in a -1.25% incremental change in high-speed DC voltage level.
22-21	PHY_PC_TXRISETUNE	HS Transmitter Rise/Fall TIme Adjustment. Adjusts the rise/fall times of the high-speed waveform. • +1 = results in a -4% incremental change in the HS rise/fall time. • -1 = results in a +4% incremental change in the HS rise/fall time.
20-19	PHY_PC_TXRESTUNE	USB Source Impedance Adjustment. Some applications require additional devices to be added on the USB, such as a series switch, which can add significant series resistance. This bus adjusts the driver source impedance to compensate for added series resistance on the USB.
18	PHY_PC_ TXPREEMPPULSETUNE	HS Transmitter Pre-Emphasis Duration Control. Controls the duration for which the HS pre-emphasis current is sourced onto DP or DM. It is defined in terms of unit amounts. One unit of pre-emphasis duration is approximately 580 ps and is defined as 1x pre-emphasis duration. This signal valid only if either txpreempamptune[1] or txpreempamptune[0] is set to 1. • 1 = 1x, short pre-emphasis current duration. • 0 = 2x, long pre-emphasis current duration.

Table 8-54. USB_PHY_CTL2 Register Field Descriptions (continued)

Bit	Field	Description
17-16	PHY_PC_TXPREEMPAMPTUNE	HS Transmitter Pre-Emphasis Current Control. Controls the amount of current sourced to DP and DM after a J-to-K or K-to-J transition. The HS Transmitter pre-emphasis current is defined in terms of unit amounts. One unit amount is approximately 600 µ;A and is defined as 1x pre-emphasis current. • 11 = 3x pre-emphasis current. • 10 = 2x pre-emphasis current. • 01 = 1x pre-emphasis current. • 00 = HS Transmitter pre-emphasis is disabled.
15-14	PHY_PC_TXHSXVTUNE	Transmitter High-Speed Crossover Adjustment. Adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. • 11 = Default setting. • 10 = +15 mV • 01 = -15 mV • 00 = Reserved
13-10	PHY_PC_TXFSLSTUNE	FS/LS Source Impedance Adjustment. Adjusts the low- and full-speed single-ended source impedance while driving high. This parameter control is encoded in thermometer code. • +1 = results in a -2.5% incremental change in threshold voltage level. • -1 = results in a +2.5% incremental change in threshold voltage level. Any non-thermometer code setting (that is 1001) is not supported and reserved.
9-7	PHY_PC_SQRXTUNE	Squelch Threshold Adjustment. Adjusts the voltage level for the threshold used to detect valid high-speed data. • +1 = results in a -5% incremental change in threshold voltage level. • -1 = results in a +5% incremental change in threshold voltage level.
6-4	PHY_PC_OTGTUNE	VBUS Valid Threshold Adjustment. Adjusts the voltage level for the VBUS valid threshold. • +1 = results in a +1.5% incremental change in threshold voltage level. • -1 = results in a -1.5% incremental change in threshold voltage level.
3	Reserved	Reserved
2-0	PHY_PC_COMPDISTUNE	Disconnect Threshold Adjustment. Adjusts the voltage level for the threshold used to detect a disconnect event at the host. • +1 = results in a +1.5% incremental change in the threshold voltage level. • -1 = results in a -1.5% incremental change in the threshold voltage level.

Figure 8-37. USB_PHY_CTL3 Register

31	30	29	23
Reserved		PHY_PC_PCS_TX_SWING_FULL	
R-0		R/W-1111000	

22	17	16	11	10	5	4	0
PHY_PC_PCS_TX_DEEMPH_6DB		Reserved		PHY_PC_PCS_TX_DEEMPH_3P5DB		PHY_PC_LOS_LEVEL	
R/W-100000		R-0		R/W-010101		R/W-01001	

Legend: R = Read only; R/W = Read/Write, -*n* = value after reset

Table 8-55. USB_PHY_CTL3 Register Field Descriptions

Bit	Field	Description
31-30	Reserved	Reserved
29-23	PHY_PC_PCS_TX_SWING_FULL	Tx Amplitude (Full Swing Mode). Sets the launch amplitude of the transmitter. It can be used to tune Rx eye for compliance.

Table 8-55. USB_PHY_CTL3 Register Field Descriptions (continued)

Bit	Field	Description
22-17	PHY_PC_PCS_TX_DEEMPH_6DB	Tx De-Emphasis at 6 dB. Sets the Tx driver de-emphasis value when pipeP_tx_deemph[1:0] is set to 10b (according to the PIPE3 specification). This bus is provided for completeness and as a second potential launch amplitude.
16-11	Reserved	Reserved
10-5	PHY_PC_PCS_TX_DEEMPH_3P5DB	Tx De-Emphasis at 3.5 dB. Sets the Tx driver de-emphasis value when pipeP_tx_deemph[1:0] is set to 10b (according to the PIPE3 specification). Can be used for Rx eye compliance.
4-0	PHY_PC_LOS_LEVEL	Loss-of-Signal Detector Sensitivity Level Control. Sets the LOS detection threshold level. This signal must be set to 0x9.

Figure 8-38. USB_PHY_CTL4 Register

31	30	29	28
PHY_SSC_EN	PHY_REF_USE_PAD	PHY_REF_SSP_EN	PHY_MPLL_REFSSC_CLK_EN
R/W-1	R/W-0	R/W-0	R/W-0

27 22	21	20 19	18	17
PHY_FSEL	PHY_RETENABLEN	PHY_REFCLKSEL	PHY_COMMONONN	Reserved
R/W-100111	R/W-1	R/W-10	R/W-0	R-0

16	15	14 12	11 7	6 0
PHY_OTG_VBUSVLDEXTSEL	PHY_OTG_OTGDISABLE	PHY_PC_TX_VBOOST_LVL	PHY_PC_LANE0_TX_TERM_OFFSET	Reserved
R/W-0	R/W-1	R/W-100	R/W-00000	R-0

Legend: R = Read only; R/W = Read/Write, -*n* = value after reset

Table 8-56. USB_PHY_CTL4 Register Field Descriptions

Bit	Field	Description
31	PHY_SSC_EN	Spread Spectrum Enable. Enables spread spectrum clock production (0.5% down-spread at ~31.5 KHz) in the USB3.0 PHY. If the reference clock already has spread spectrum applied, ssc_en must be de-asserted.
30	PHY_REF_USE_PAD	Select Reference Clock Connected to ref_pad_clk_{p,m}. When asserted, selects the external ref_pad_clk_{p,m} inputs as the reference clock source. When de-asserted, ref_alt_clk_{p,m} are selected for an on-chip reference clock source.
29	PHY_REF_SSP_EN	Reference Clock Enables for SS function. Enables the reference clock to the prescaler. The ref_ssp_en signal must remain de asserted until the reference clock is running at the appropriate frequency, at which point ref_ssp_en can be asserted. For lower power states, ref_ssp_en can also be de asserted.
28	PHY_MPLL_REFSSC_CLK_EN	Double-Word Clock Enable. Enables/disables the mpll_refssc_clk signal. To prevent clock glitch, it must be changed when the PHY is inactive.
27-22	PHY_FSEL	Frequency Selection. Selects the reference clock frequency used for both SS and HS operations. The value for fsel combined with the other clock and enable signals will determine the clock frequency used for SS and HS operations and if a shared or separate reference clock will be used.
21	PHY_RETENABLEN	Lowered Digital Supply Indicator. Indicates that the vp digital power supply has been lowered in Suspend mode. This signal must be de-asserted before the digital power supply is lowered. • 1 = Normal operating mode. • 0 = The analog blocks are powered down.

Table 8-56. USB_PHY_CTL4 Register Field Descriptions (continued)

Bit	Field	Description
20-19	PHY_REFCLKSEL	Reference Clock Select for PLL Block. Selects reference clock source for the HS PLL block. • 11 = HS PLL uses EXTREFCLK as reference. • 10 = HS PLL uses either ref_pad_clk_{p,m} or ref_alt_clk_{p,m} as reference. • x0 = Reserved.
18	PHY_COMMONONN	Common Block Power-Down Control. Controls the power-down signals in the HS Bias and PLL blocks when the USB3.0 PHY is in Suspend or Sleep mode. • 1 = In Suspend or Sleep mode, the HS Bias and PLL blocks are powered down. • 0 = In Suspend or Sleep mode, the HS Bias and PLL blocks remain powered and continue to draw current.
17	Reserved	Reserved
16	PHY_OTG_VBUSVLDEXTSEL	External VBUS Valid Select. Selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid. • 1 = VBUSVLDEXT input is used. • 0 = Internal Session Valid comparator is used.
15	PHY_OTG_OTGDISABLE	OTG Block Disable. Powers down the OTG block, which disables the VBUS Valid and Session End comparators. The Session Valid comparator (the output of which is used to enable the pull-up resistor on DP in Device mode) is always on irrespective of the state of otgdisable. If the application does not use the OTG function, setting this signal to high to save power. • 1 = OTG block is powered down. • 0 = OTG block is powered up.
14-12	PHY_PC_TX_VBOOST_LVL	Tx Voltage Boost Level. Sets the boosted transmit launch amplitude (mV_{ppd}). The default setting is intended to set the launch amplitude to approximately 1,008mV_{ppd}. • +1 = results in a +156 mV_{ppd} change in the Tx launch amplitude. • -1 = results in a -156 mV_{ppd} change in the Tx launch amplitude.
11-7	PHY_PC_LANE0_TX_TERM_OFFSET	Transmitter Termination Offset. Enables adjusting the transmitter termination value from the default value of 60 Ω.
6-0	Reserved	Reserved

Figure 8-39. USB_PHY_CTL5 Register

31	21	20	19	13
Reserved		PHY_REF_CLKDIV2	PHY_MPLL_MULTIPLIER[6:0]	
R-0		R/W-0	R/W +0011001	

12	4	3	2	0
PHY_SSC_REF_CLK_SEL		Reserved	PHY_SSC_RANGE	
R/W-000000000		R-0	R/W-000	

Legend: R = Read only; R/W = Read/Write, -*n* = value after reset

Table 8-57. USB_PHY_CTL5 Register Field Descriptions

Bit	Field	Description
31-21	Reserved	Reserved
20	PHY_REF_CLKDIV2	Input Reference Clock Divider Control. If the input reference clock frequency is greater than 100 MHz, this signal must be asserted. The reference clock frequency is then divided by 2 to keep it in the range required by the MPLL. When this input is asserted, the ref_ana_usb2_clk (if used) frequency will be the reference clock frequency divided by 4.
19-13	PHY_MPLL_MULTIPLIER[6:0]	MPLL Frequency Multiplier Control. Multiplies the reference clock to a frequency suitable for intended operating speed.
12-4	PHY_SSC_REF_CLK_SEL	Spread Spectrum Reference Clock Shifting. Enables non-standard oscillator frequencies to generate targeted MPLL output rates. Input corresponds to frequency-synthesis coefficient. • . ssc_ref_clk_sel[8:6] = modulous - 1 • . ssc_ref_clk_sel[5:0] = 2's complement push amount.
3	Reserved	Reserved
2-0	PHY_SSC_RANGE	Spread Spectrum Clock Range. Selects the range of spread spectrum modulation when ssc_en is asserted and the PHY is spreading the high-speed transmit clocks. Applies a fixed offset to the phase accumulator.

9 Device Operating Conditions

9.1 Absolute Maximum Ratings(1)

Over Operating Case Temperature Range (Unless Otherwise Noted)

Supply voltage range(2):	CVDD	-0.3 V to 1.3 V
	CVDD1	-0.3 V to 1.3 V
	DVDD15	-0.3 V to 1.98 V
	DVDD18	-0.3 V to 2.45 V
	DDR3VREFSSTL	0.49 × DVDD15 to 0.51 × DVDD15
	VDDAHV	-0.3 V to 1.98 V
	VDDALV	-0.3 V to 0.935 V
	USB0DVDD33, USB0DVDD33	-0.3V to 3.63 V
	VDDUSB0, VDDUSB1	-0.3V to 0.935 V
	USB0VP, USB1VP	-0.3V to 0.935 V
	USB0VPH, USB1VPH	-0.3V to 3.63 V
	USB0VPTX, USB1VPTX	-0.3V to 0.935 V
	AVDDA1, AVDDA2, AVDDA3	-0.3 V to 1.98 V
	AVDDA6, AVDDA7 AVDDA8, AVDDA9, AVDDA10	-0.3 V to 1.98 V
	VSS Ground	0 V
Input voltage (V_I) range(3):	LVCMOS (1.8 V)	-0.3 V to DVDD18+0.3 V
	DDR3	-0.3 V to 1.98 V
	I^2C	-0.3 V to 2.45 V
	LVDS	-0.3 V to DVDD18+0.3 V
	LJCB	-0.3 V to 1.3 V
	SerDes	-0.3 V to VDDAHV1+0.3 V
Output voltage (V_O) range(3):	LVCMOS (1.8 V)	-0.3 V to DVDD18+0.3 V
	DDR3	-0.3 V to 1.98 V
	I^2C	-0.3 V to 2.45 V
	SerDes	-0.3 V to VDDAHV+0.3 V
Operating case temperature range, T_C:	Commercial	0°C to 85°C
	Extended	-40°C to 100°C
ESD stress voltage, V_{ESD}(4)	HBM (human body model)(5)	±1000 V
	CDM (charged device model)(6)	±250 V
Overshoot/undershoot(7)	LVCMOS (1.8 V)	20% overshoot/undershoot for 20% of signal duty cycle
	DDR3	
	I^2C	
Storage temperature range, T_{stg}:		-65°C to 150°C

(1) Stresses beyond those listed under *absolute maximum ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *recommended operating conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
(2) All voltage values are with respect to V_{SS}.
(3) For USB High-Speed, Full-Speed, and Low -Speed modes, USB I/Os adhere to Universal Serial Bus, revision 2.0 standard. For USB Super-Speed mode, USB I/Os adhere to Universal Serial Bus, revision 3.1 specification, revision 1.0 standard.
(4) Electrostatic discharge (ESD) to measure device sensitivity/immunity to damage caused by electrostatic discharges into the device.
(5) Level listed above is the passing level per ANSI/ESDA/JEDEC JS-001-2010. JEDEC document JEP155 states that 500 V HBM allows safe manufacturing with a standard ESD control process, and manufacturing with less than 500 V HBM is possible if necessary precautions are taken. Pins listed as 1000 V may actually have higher performance.
(6) Level listed above is the passing level per EIA-JEDEC JESD22-C101E. JEDEC document JEP157 states that 250 V CDM allows safe manufacturing with a standard ESD control process. Pins listed as 250 V may actually have higher performance.
(7) Overshoot/Undershoot percentage relative to I/O operating values - for example the maximum overshoot value for 1.8 V LVCMOS signals is DVDD18 + 0.20 × DVDD18 and maximum undershoot value would be V_{SS} - 0.20 × DVDD18

9.2 Recommended Operating Conditions[1][2]

			MIN	NOM	MAX	UNIT
CVDD	SR core supply	Initial[3]	1.0	1.05	1.10	V
		1250MHz and 1400MHz Device	SRVnom*0.95[4]	SRVnom	SRVnom*1.05	V
CVDD1	Core supply		0.95	1.0	1.05	V
DVDD18	1.8-V supply I/O voltage		1.71	1.8	1.89	V
DVDD15	DDR3 I/O voltage	DDR3	1.425	1.5	1.575	V
		DDR3L @ 1.5 V	1.425	1.5	1.575	
		DDR3L @ 1.35 V	1.283	1.35	1.45	
DDR3VREFSSTL	DDR3 reference voltage		0.49 × DVDD15	0.5 × DVDD15	0.51 × DVDD15	V
VDDAHV	SerDes regulator supply		1.71	1.8	1.89	V
VDDALH	SerDes termination supply		0.807	0.85	0.892	V
AVDDx[5]	PLL analog, DDR DLL supply		1.71	1.8	1.89	V
USB0VP, USB1VP	0.85-V USB PHY supply		0.807	0.85	0.892	V
USB0VPH, USB1VPH	3.3-V USB		3.135	3.3	3.465	V
USB0VPTX, USB1VPTX	USBPHY Transmit supply		0.807	0.85	0.892	V
VDDUSB0, VDDUSB1	USB PHY supply		0.807	0.85	0.892	V
USB0DVDD33, USB1DVDD33	USB 3.3-V high supply		3.135	3.3	3.465	V
V_{SS}	Ground		0	0	0	V
V_{IH}[6]	High-level input voltage	LVCMOS (1.8 V)	0.65 × DVDD18			V
		I^2C	0.7 × DVDD18			V
		DDR3 EMIF	VREFSSTL + 0.1			V
V_{IL}[6]	Low-level input voltage	LVCMOS (1.8 V)			0.35 × DVDD18	V
		DDR3 EMIF	-0.3		VREFSSTL - 0.1	V
		I^2C			0.3 × DVDD18	V
T_C	Operating case temperature	Commercial	0		85	°C
		Extended	-40		100	°C

(1) All differential clock inputs comply with the LVDS Electrical Specification, IEEE 1596.3-1996 and all SerDes I/Os comply with the XAUI Electrical Specification, IEEE 802.3ae-2002.
(2) All SerDes I/Os comply with the XAUI Electrical Specification, IEEE 802.3ae-2002.
(3) Users are required to program their board CVDD supply initial value to 1.0 V on the device. The initial CVDD voltage at power-on will be 1.0V nominal and it must transition to VID set value, immediately after being presented on the VCNTL pins. This is required to maintain full power functionality and reliability targets guaranteed by TI.
(4) SRVnom refers to the unique SmartReflex core supply voltage that has a potential range of 0.8 V and 1.1 V which preset from the factory for each individual device. Your device may never be programmed to operate at the upper range but has been designed accordingly should it be determined to be acceptable or necessary. Power supplies intended to support the variable SRV function shall be capable of providing a 0.8V-1.1V dynamic range using a 4- or 6-bit binary input value which as provided by the SOC SmartReflex output.
(5) Where x=1,2,3,4... to indicate all supplies of the same kind.
(6) For USB High-Speed, Full-Speed, and Low -Speed modes, USB I/Os adhere to Universal Serial Bus, revision 2.0 standard. For USB Super-Speed mode, USB I/Os adhere to Universal Serial Bus, revision 3.1 specification, revision 1.0 standard.

9.3 Electrical Characteristics

Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted)

PARAMETER			TEST CONDITIONS(1)	MIN	TYP	MAX	UNIT
V_{OH}(2)	High-level output voltage	LVCMOS (1.8 V)	$I_O = I_{OH}$	DVDD18 - 0.45			V
		DDR3		DVDD15 - 0.4			
		I^2C(3)			(3)		
V_{OL}(2)	Low-level output voltage	LVCMOS (1.8 V)	$I_O = I_{OL}$			0.45	V
		DDR3				0.4	
		I^2C	I_O = 3 mA, pulled up to 1.8 V			0.4	
I_I(4)	Input current [DC]	LVCMOS (1.8 V)	No IPD/IPU	-10		10	µA
			Internal pullup	50	100	170	
			Internal pulldown	-170	-100	-50	
		I^2C	0.1 × DVDD18 V < V_I < 0.9 × DVDD18 V	-10		10	µA
I_{OH}	High-level output current [DC]	LVCMOS (1.8 V)				-6	mA
		DDR3				-8	
		I^2C(5)			(5)		
I_{OL}	Low-level output current [DC]	LVCMOS (1.8 V)				6	mA
		DDR3				8	
		I^2C				3	
I_{OZ}(6)	Off-state output current [DC]	LVCMOS (1.8 V)		-10		10	µA
		DDR3		-10		10	
		I^2C		-10		10	

(1) For test conditions shown as MIN, MAX, or TYP, use the appropriate value specified in the recommended operating conditions table.
(2) For USB High-Speed, Full-Speed, and Low -Speed modes, USB I/Os adhere to Universal Serial Bus, revision 2.0 standard. For USB Super-Speed mode, USB I/Os adhere to Universal Serial Bus, revision 3.1 specification, revision 1.0 standard.
(3) I^2C uses open collector IOs and does not have a V_{OH} Minimum.
(4) I_I applies to input-only pins and bidirectional pins. For input-only pins, I_I indicates the input leakage current. For bidirectional pins, I_I includes input leakage current and off-state (Hi-Z) output leakage current.
(5) I^2C uses open collector IOs and does not have a I_{OH} Maximum.
(6) I_{OZ} applies to output-only pins, indicating off-state (Hi-Z) output leakage current.

9.4 Power Supply to Peripheral I/O Mapping

Table 9-1. Power Supply to Peripheral I/O Mapping(1)(2)

Over Recommended Ranges of Supply Voltage and Operating Case Temperature (Unless Otherwise Noted)

POWER SUPPLY		I/O BUFFER TYPE	ASSOCIATED PERIPHERAL
CVDD	Supply core AVS voltage	LJCB	CORECLK(P|N) PLL input buffer
			DDR3CLK(P|N) PLL input buffer
			NETCPCLK(P|N) PLL input buffer
			USBCLK(P|M) PLL input buffer
			SGMII0CLK(P|N) PLL input buffer
VDDALV		SerDes/CML	SERDES low voltage
VDDAHV	SerDes IO voltage	SerDes/CML	PCIECLK(P|N) SerDes Clock Reference
			HYPCLK(P|N) SerDes Clock Reference
			XFICLK(P|N) SerDes Clock Reference(3)
DVDD15	DDR3 memory I/O voltage	DDR3 (1.5/1.35 V)	All DDR3 memory controller peripheral I/O buffer
DVDD18	1.8-V supply I/O voltage	LVCMOS (1.8 V)	All GPIO peripheral I/O buffer
			All JTAG and EMU peripheral I/O buffer
			All TIMER peripheral I/O buffer
			All SPI peripheral I/O buffer
			All TSIP peripheral I/O buffer
			All RESETs, control peripheral I/O buffer
			All SmartReflex peripheral I/O buffer
			All Hyperlink sideband peripheral I/O buffer
			All MDIO peripheral I/O buffer
			All UART peripheral I/O buffer
		Open-drain (1.8 V)	All I^2C peripheral I/O buffer
		LVDS	TSREFCLK SerDes Clock Reference

(1) Please note that this table does not attempt to describe all functions of all power supply terminals but only those whose purpose it is to power peripheral I/O buffers and clock input buffers.
(2) Please see the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0) for more information about individual peripheral I/O.
(3) 10 GbE supported in AM5K2E04 only.

10 AM5K2E0x Peripheral Information and Electrical Specifications

This chapter covers the various peripherals on the AM5K2E0x device. Peripheral-specific information, timing diagrams, electrical specifications, and register memory maps are described in this chapter.

10.1 Recommended Clock and Control Signal Transition Behavior

All clocks and control signals ***must*** transition between V_{IH} and V_{IL} (or between V_{IL} and V_{IH}) in a monotonic manner.

10.2 Power Supplies

The following sections describe the proper power-supply sequencing and timing needed to properly power on the AM5K2E0x. The various power supply rails and their primary functions are listed in Table 10-1.

Table 10-1. Power Supply Rails on the AM5K2E0x

NAME	PRIMARY FUNCTION	VOLTAGE	NOTES
AVDDAx	Core PLL, DDR3 DLL supply voltage	1.8 V	Core PLL, DDR3 DLL supply
DVDD15	DDR3 I/O power supply voltage	1.5/1.35 V	DDR3 I/O power supply
DVDD18	1.8-V I/O power supply voltage	1.8 V	1.8-V I/O power supply
USB0DVDD33, USB1DVDD33	USB 3.3-V IO supply	3.3 V	USB high voltage supply
VDDAHV	SerDes I/O power supply voltage	1.8 V	SerDes I/O power supply
VDDALV	SerDes analog power supply voltage	0.85 V	SerDes analog supply
VDDUSB0, VDDUSB1	USB LV PHY power supply voltage	0.85 V	USB LV PHY supply
USB0VP, USB0VPTX, USB0VP, USB0VPTX	Filtered 0.85-V supply voltage	0.85 V	Filtered 0.85-V USB supply
VSS	Ground	GND	Ground

10.2.1 Power-Up Sequencing

This section defines the requirements for a power-up sequencing from a power-on reset condition. There are two acceptable power sequences for the device.

The first sequence stipulates the **core voltages starting before the IO voltages** as shown below.

1. CVDD
2. CVDD1, VDDAHV, AVDDAx, DVDD18
3. DVDD15
4. VDDALV, VDDUSBx, USBxVP, USBxVPTX
5. USBxDVDD33

The second sequence provides compatibility with other TI processors with the **IO voltage starting before the core voltages** as shown below.

1. VDDAHV, AVDDAx, DVDD18
2. CVDD
3. CVDD1
4. DVDD15
5. VDDALV, VDDUSBx, USBxVP, USBxVPTX
6. USBxDVDD33

The clock input buffers for CORECLK, DDRCLK, NETCPCLK, and SGMIICLK use CVDD as a supply voltage. These clock inputs are not failsafe and must be held in a high-impedance state until CVDD is at a valid voltage level. Driving these clock inputs high before CVDD is valid could cause damage to the device. Once CVDD is valid, it is acceptable that the P and N legs of these clocks may be held in a static state (either high and low or low and high) until a valid clock frequency is needed at that input. To avoid internal oscillation, the clock inputs should be removed from the high impedance state shortly after CVDD is present.

If a clock input is not used, it must be held in a static state. To accomplish this, the N leg should be pulled to ground through a 1-kΩ resistor. The P leg should be tied to CVDD to ensure it will not have any voltage present until CVDD is active. Connections to the IO cells powered by DVDD18 and DVDD15 are not failsafe and should not be driven high before these voltages are active. Driving these IO cells high before DVDD18 or DVDD15 are valid could cause damage to the device.

The device initialization is divided into two phases. The first phase consists of the time period from the activation of the first power supply until the point at which all supplies are active and at a valid voltage level. Either of the sequencing scenarios described above can be implemented during this phase. The figures below show both the core-before-IO voltage sequence and the IO-before-core voltage sequence. POR must be held low for the entire power stabilization phase.

This is followed by the device initialization phase. The rising edge of POR followed by the rising edge of RESETFULL triggers the end of the initialization phase, but both must be inactive for the initialization to complete. POR must always go inactive before RESETFULL goes inactive as described below. SYSCLK1 in the following section refers to the clock that is used by the CorePacs. See Figure 10-7 for more details.

10.2.1.1 Core-Before-IO Power Sequencing

The details of the Core-before-IO power sequencing are defined in Table 10-2. Figure 10-1 shows power sequencing and reset control of the AM5K2E0x. POR may be removed after the power has been stable for the required 100 µsec. RESETFULL must be held low for a period (see item 9 in Figure 10-1) after the rising edge of POR, but may be held low for longer periods if necessary. The configuration bits shared with the GPIO pins will be latched on the rising edge of RESETFULL and must meet the setup and hold times specified. SYSCLK1 must always be active before POR can be removed.

NOTE

TI recommends a maximum of 80 ms between one power rail being valid and the next power rail in the sequence starting to ramp.

Table 10-2. Core-Before-IO Power Sequencing

ITEM	SYSTEM STATE
1	**Begin Power Stabilization Phase** • CVDD (core AVS) ramps up. • POR must be held low through the power stabilization phase. Because POR is low, all the core logic that has asynchronous reset (created from POR) is put into the reset state. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
2a	• CVDD1 (core constant) ramps at the same time or within 80 ms of CVDD. Although ramping CVDD1 simultaneously with CVDD is permitted, the voltage for CVDD1 **must never exceed** CVDD until after CVDD has reached a valid voltage. • The purpose of ramping up the core supplies close to each other is to reduce crowbar current. CVDD1 should trail CVDD as this will ensure that the Word Lines (WLs) in the memories are turned off and there is no current through the memory bit cells. If, however, CVDD1 (core constant) ramps up before CVDD (core AVS), then the worst-case current could be on the order of twice the specified draw of CVDD1. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less. • The timing for CVDD1 is based on CVDD valid. CVDD1 and DVDD18/ADDAVH/AVDDAx may be enabled at the same time but do not need to ramp simultaneously. CVDD1 may be valid before or after DVDD18/ADDAVH/AVDDAx are valid, as long as the timing above is met.

Table 10-2. Core-Before-IO Power Sequencing (continued)

ITEM	SYSTEM STATE
2b	• VDDAHV, AVDDAx and DVDD18 ramp at the same time or shortly following CVDD. DVDD18 must be enabled within 80 ms of CVDD valid and must ramp monotonically and reach a stable level in 20ms or less. **This results in no more than 100 ms from the time when CVDD is valid to the time when DVDD18 is valid.** • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less. • The timing for DVDD18/ADDAVH/AVDDAx is based on CVDD valid. DVDD18/ADDAVH/AVDDAx and CVDD1 may be enabled at the same time but do not need to ramp simultaneously. DVDD18/ADDAVH/AVDDAx may be valid before or after CVDD1 is valid, as long as the timing above is met.
2c	• Once CVDD is valid, the clock drivers can be enabled. Although the clock inputs are not necessary at this time, they should either be driven with a valid clock or be held in a static state with one leg high and one leg low.
2d	• The DDRCLK and SYSCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by item 7.
3	• DVDD15 can ramp up within 80ms of when DVDD18 is valid. • $\overline{\text{RESETSTAT}}$ is driven low once the DVDD18 supply is available. • All LVCMOS input and bidirectional pins must not be driven or pulled high until DVDD18 is present. Driving an input or bidirectional pin before DVDD18 is valid could cause damage to the device. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
3a	• $\overline{\text{RESET}}$ may be driven high any time after DVDD18 is at a valid level. $\overline{\text{RESET}}$ must be high before $\overline{\text{POR}}$ is driven high.
4	• VDDALV, VDDUSBx, USBxVP and USBxVPTX ramp up within 80ms of when DVDD15 is valid. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
5	• USBxDVDD33 supply is ramped up within 80 ms of when VDDALV, VDDUSBx, USBxVP and USBxVPTX are valid. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
6	• $\overline{\text{POR}}$ must continue to remain low for at least 100 µs after all power rails have stabilized. **End power stabilization phase**
7	• Device initialization requires 500 SYSCLK1 periods after the Power Stabilization Phase. The maximum clock period is 33.33 nsec, so a delay of an additional 16 µs is required before a rising edge of $\overline{\text{POR}}$. The clock must be active during the entire 16 µs.
8	• $\overline{\text{RESETFULL}}$ must be held low for at least 24 transitions of the SYSCLK1 after $\overline{\text{POR}}$ has stabilized at a high level.
9	• The rising edge of the $\overline{\text{RESETFULL}}$ will remove the reset to the eFuse farm allowing the scan to begin. • Once device initialization and the eFuse farm scan are complete, the $\overline{\text{RESETSTAT}}$ signal is driven high. This delay will be 10000 to 50000 clock cycles. **End device initialization phase**
10	• GPIO configuration bits must be valid for at least 12 transitions of the SYSCLK1 before the rising edge of $\overline{\text{RESETFULL}}$.
11	• GPIO configuration bits must be held valid for at least 12 transitions of the SYSCLK1 after the rising edge of $\overline{\text{RESETFULL}}$.

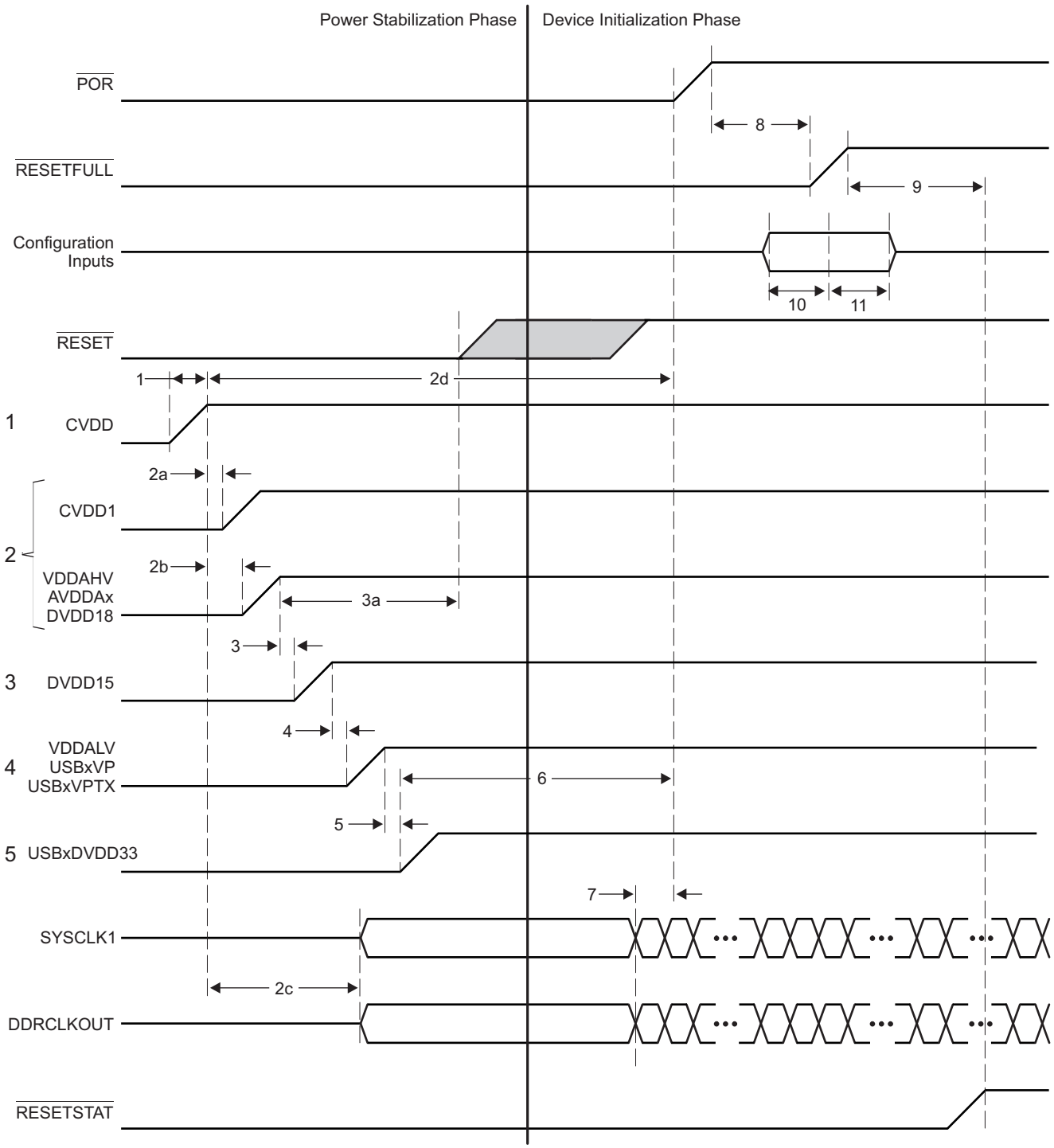


Figure 10-1. Core-Before-IO Power Sequencing

10.2.1.2 IO-Before-Core Power Sequencing

The timing diagram for IO-before-core power sequencing is shown in Figure 10-2 and defined in Table 10-3.

NOTE

TI recommends a maximum of 100 ms between one power rail being valid, and the next power rail in the sequence starting to ramp.

Table 10-3. IO-Before-Core Power Sequencing

ITEM	SYSTEM STATE
1	**Begin Power Stabilization Phase** • VDDAHV, AVDDAx and DVDD18 ramp up. • $\overline{\text{POR}}$ must be held low through the power stabilization phase. Because $\overline{\text{POR}}$ is low, all the core logic that has asynchronous reset (created from $\overline{\text{POR}}$) is put into the reset state. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
2	• CVDD (core AVS) ramps within 80 ms from the time ADDAHV, AVDDAx and DVDD18 are valid. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
2a	• $\overline{\text{RESET}}$ may be driven high any time after DVDD18 is at a valid level. must be high before $\overline{\text{POR}}$ is driven high.
3	• CVDD1 (core constant) ramp at the same time or within 80 ms following CVDD. Although ramping CVDD1 simultaneously with CVDD is permitted, the voltage for CVDD1 **must never exceed** CVDD until after CVDD has reached a valid voltage. • The purpose of ramping up the core supplies close to each other is to reduce crowbar current. CVDD1 should trail CVDD as this will ensure that the Word Lines (WLs) in the memories are turned off and there is no current through the memory bit cells. If, however, CVDD1 (core constant) ramp up before CVDD (core AVS), then the worst-case current could be on the order of twice the specified draw of CVDD1. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
3a	• Once CVDD is valid, the clock drivers can be enabled. Although the clock inputs are not necessary at this time, they should either be driven with a valid clock or held in a static state.
3b	• The DDRCLK and SYSCLK1 may begin to toggle anytime between when CVDD is at a valid level and the setup time before $\overline{\text{POR}}$ goes high specified by item 8.
4	• DVDD15 can ramp up within 80 ms of when CVDD1 is valid. • $\overline{\text{RESETSTAT}}$ is driven low once the DVDD18 supply is available. • All LVCMOS input and bidirectional pins must not be driven or pulled high until DVDD18 is present. Driving an input or bidirectional pin before DVDD18 is valid could cause damage to the device. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
5	• VDDALV, VDDUSBx, USBxVP and USBxVPTX should ramp up within 80 ms of when DVDD15 is valid. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
6	• USBxDVDD33 supply is ramped up within 80 ms of when VDDALV, VDDUSBx, USBxVP and USBxVPTX are valid. • Each supply must ramp monotonically and must reach a stable valid level in 20 ms or less.
7	• $\overline{\text{POR}}$ must continue to remain low for at least 100 µs after all power rails have stabilized. **End power stabilization phase**
8	• Device initialization requires 500 SYSCLK1 periods after the Power Stabilization Phase. The maximum clock period is 33.33 nsec, so a delay of an additional 16 µs is required before a rising edge of $\overline{\text{POR}}$. The clock must be active during the entire 16 µs.
9	• $\overline{\text{RESETFULL}}$ must be held low for at least 24 transitions of the SYSCLK1 after $\overline{\text{POR}}$ has stabilized at a high level.
10	• The rising edge of the $\overline{\text{RESETFULL}}$ will remove the reset to the efuse farm allowing the scan to begin. • Once device initialization and the efuse farm scan are complete, the $\overline{\text{RESETSTAT}}$ signal is driven high. This delay will be 10000 to 50000 clock cycles. **End device initialization phase**
11	• GPIO configuration bits must be valid for at least 12 transitions of the SYSCLK1 before the rising edge of $\overline{\text{RESETFULL}}$.
12	• GPIO configuration bits must be held valid for at least 12 transitions of the SYSCLK1 after the rising edge of $\overline{\text{RESETFULL}}$.

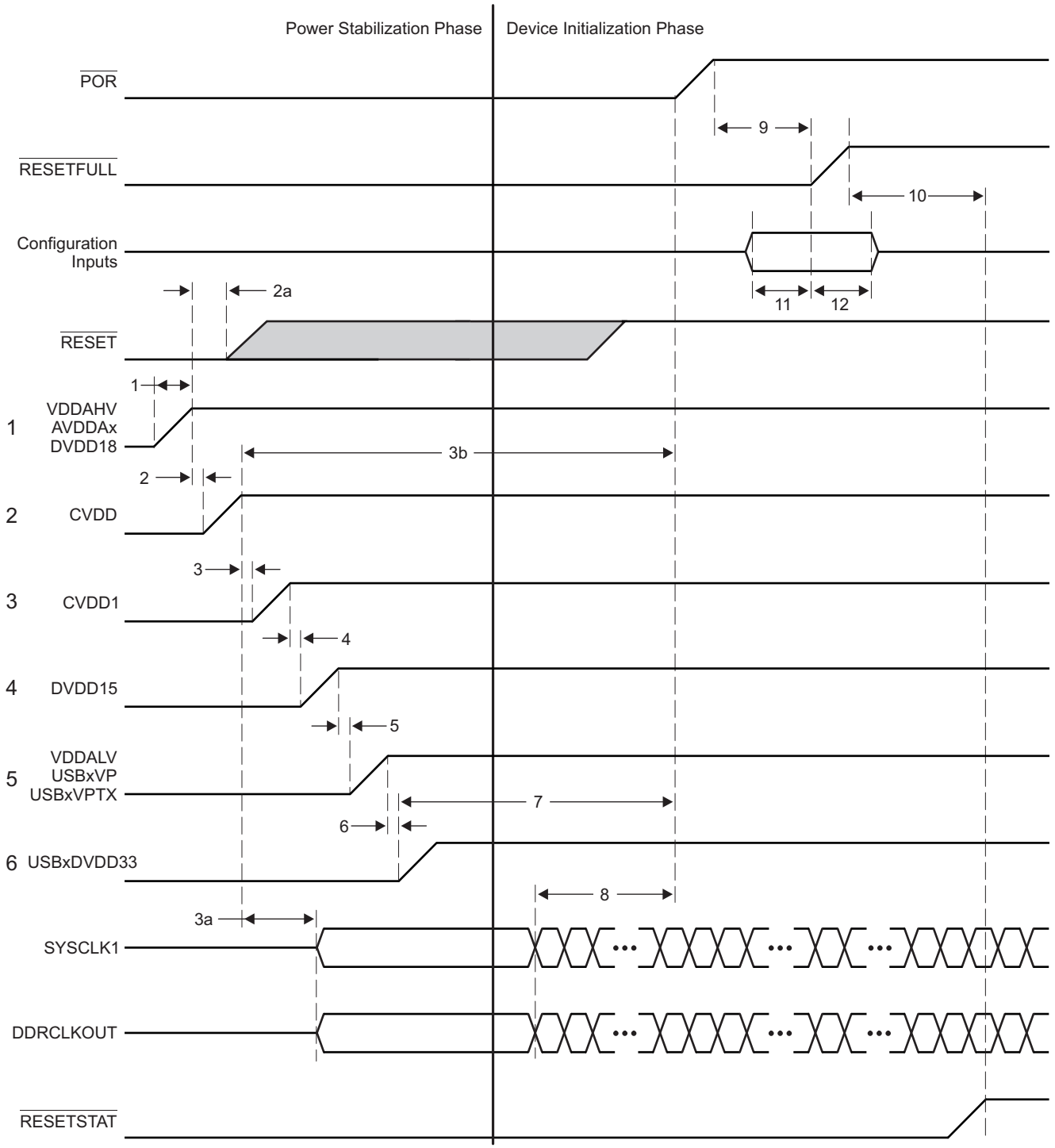


Figure 10-2. IO-Before-Core Power Sequencing

10.2.1.3 Prolonged Resets

Holding the device in $\overline{\text{POR}}$, $\overline{\text{RESETFULL}}$, or $\overline{\text{RESET}}$ for long periods of time may affect the long-term reliability of the part (due to an elevated voltage condition that can stress the part). The device should not be held in a reset for times exceeding one hour at a time and no more than 5% of the total lifetime for which the device is powered-up. Exceeding these limits will cause a gradual reduction in the reliability of the part. This can be avoided by allowing the device to boot and then configuring it to enter a hibernation state soon after power is applied. This will satisfy the reset requirement while limiting the power consumption of the device.

10.2.1.4 Clocking During Power Sequencing

Some of the clock inputs are required to be present for the device to initialize correctly, but behavior of many of the clocks is contingent on the state of the boot configuration pins. Table 10-4 describes the clock sequencing and the conditions that affect clock operation. Note that all clock drivers should be in a high-impedance state until CVDD is at a valid level and that all clock inputs be either active or in a static state with one leg pulled to ground and the other connected to CVDD.

Table 10-4. Clock Sequencing

CLOCK	CONDITION	SEQUENCING
DDRCLK	None	Must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
CORECLK	None	CORECLK is used to clock the core PLL. It must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
NETCPCLK	NETCPCLKSEL = 0	NETCPCLK is not used and should be tied to a static state.
	NETCPCLKSEL = 1	NETCPCLK is used as a source for the NETCP PLL. It must be present before the NETCP PLL is removed from reset and programmed.
PCIECLK	PCIE will be used as a boot device.	PCIECLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	PCIE will be used after boot.	PCIECLK is used as a source to the PCIE SerDes PLL. It must be present before the PCIe is removed from reset and programmed.
	PCIE will not be used.	PCIECLK is not used and should be tied to a static state.
HYPLNK0CLK	HyperLink will be used as a boot device.	HYPLNK0CLK must be present 16 µsec before $\overline{\text{POR}}$ transitions high.
	HyperLink will be used after boot.	HYPLNK0CLK is used as a source to the HyperLink SerDes PLL. It must be present before the HyperLink is removed from reset and programmed.
	HyperLink will not be used.	HYPLNK0CLK is not used and should be tied to a static state.

10.2.2 Power-Down Sequence

The power down sequence is the exact reverse of the power-up sequence described above. The goal is to prevent an excessive amount of static current and to prevent overstress of the device. A power-good circuit that monitors all the supplies for the device should be used in all designs. If a catastrophic power supply failure occurs on any voltage rail, $\overline{\text{POR}}$ should transition to low to prevent over-current conditions that could possibly impact device reliability.

A system power monitoring solution is needed to shut down power to the board if a power supply fails. Long-term exposure to an environment in which one of the power supply voltages is no longer present will affect the reliability of the device. Holding the device in reset is not an acceptable solution because prolonged periods of time with an active reset can affect long term reliability.

10.2.3 Power Supply Decoupling and Bulk Capacitor

To properly decouple the supply planes on the PCB from system noise, decoupling and bulk capacitors are required. Bulk capacitors are used to minimize the effects of low-frequency current transients and decoupling or bypass capacitors are used to minimize higher frequency noise. For recommendations on selection of power supply decoupling and bulk capacitors see the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0).

10.2.4 SmartReflex

Increasing the device complexity increases its power consumption. With higher clock rates and increased performance comes an inevitable penalty: increasing leakage currents. Leakage currents are present in any powered circuit, independent of clock rates and usage scenarios. This static power consumption is mainly determined by transistor type and process technology. Higher clock rates also increase dynamic power, which is the power used when transistors switch. The dynamic power depends mainly on a specific usage scenario, clock rates, and I/O activity.

Texas Instruments SmartReflex technology is used to decrease both static and dynamic power consumption while maintaining the device performance. SmartReflex in the

AM5K2E0x device is a feature that allows the core voltage to be optimized based on the process corner of the device. This requires a voltage regulator for each AM5K2E0x device.

To help maximize performance and minimize power consumption of the device, SmartReflex is required to be implemented. The voltage selection can be accomplished using 4 VCNTL pins or 6 VCNTL pins (depending on power supply device being used), which are used to select the output voltage of the core voltage regulator.

For information on implementation of SmartReflex see the *Power Consumption Summary for KeyStone TCI66x Devices* application report (SPRABL4) and the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0).

Table 10-5. SmartReflex 4-Pin 6-bit VID Interface Switching Characteristics

(see Figure 10-3)

NO.	PARAMETER		MIN	MAX	UNIT
1	td(VCNTL[4:2]-VCNTL[5])	Delay time - VCNTL[4:2] valid after VCNTL[5] low		300.00	ns
2	toh(VCNTL[5]-VCNTL[4:2])	Output hold time - VCNTL[4:2] valid after VCNTL[5]	0.07	172020C (1)	ms
3	td(VCNTL[4:2]-VCNTL[5])	Delay time - VCNTL[4:2] valid after VCNTL[5] high		300.00	ns
4	toh(VCNTL[5]-VCNTL[2:0)	Output hold time - VCNTL[4:2] valid after VCNTL[5] high	0.07	172020C	ms

(1) C = 1/SYSCLK1 frequency, in ms (see Figure 10-9)

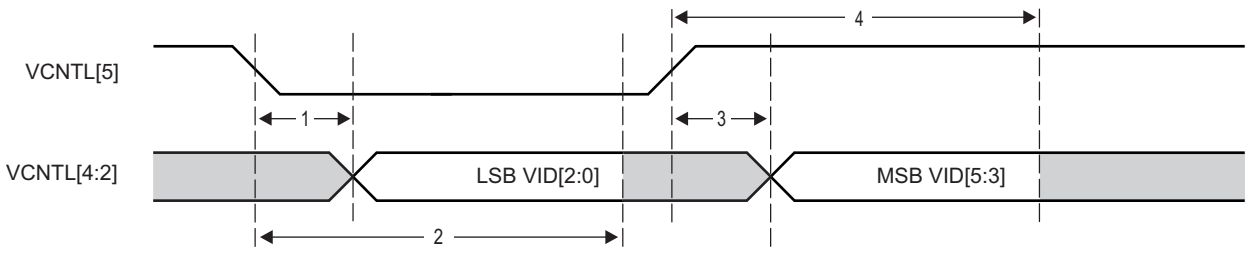


Figure 10-3. SmartReflex 4-Pin 6-Bit VID Interface Timing

10.2.5 Monitor Points

Two pairs of monitor points for the CVDD voltage level are provided. Both CVDDCMON and CVDDTMON are connected directly to the CVDD supply plane on the die itself. VSSCMON and VSSTMON are connected to the ground plane on the die. These pairs provide the best measurement points for the voltage at the silicon. They also provide the best point to connect the remote sense lines for the CVDD power supply. The use of a power supply with a differential remote sense input is highly desirable. The positive remote sense line should be connected to CVDDCMON and the negative remote sense line should be connected to VSSCMON. CVDDTMON and VSSTMON can be used as an alternative but always use either the CMON pair or the TMON pair. If the power supply remote sense is not differential CVDDCMON or CVDDTMON can be connected to the sense line.

10.3 Power Sleep Controller (PSC)

The Power Sleep Controller (PSC) includes a Global Power Sleep Controller (GPSC) and a number of Local Power Sleep Controllers (LPSC) that control overall device power by turning off unused power domains and gating off clocks to individual peripherals and modules. The PSC provides the user with an interface to control several important power and clock operations.

For information on the Power Sleep Controller, see the *KeyStone Architecture Power Sleep Controller (PSC) User's Guide* (SPRUGV4).

10.3.1 Power Domains

The device has several power domains that can be turned on for operation or off to minimize power dissipation. The Global Power Sleep Controller (GPSC) is used to control the power gating of various power domains.

The following table shows the AM5K2E0x power domains.

Table 10-6. AM66K2Ex Power Domains

DOMAIN	BLOCK(S)	NOTE	POWER CONNECTION
0	Most peripheral logic (BOOTCFG, EMIF16, I^2C, INTC, GPIO, USB)	Cannot be disabled	Always on
1	Per-core TETB and system TETB	RAMs can be powered down	Software control
2	Network Coprocessor	Logic can be powered down	Software control
3	PCIe0	Logic can be powered down	Software control
4	Reserved		
5	HyperLink	Logic can be powered down	Software control
6	SmartReflex	Cannot be disabled	Always on
7	MSMC RAM	MSMC RAM can be powered down	Software control
8	Reserved		
9	Reserved		
10	Reserved		
11	Reserved		
12	Reserved		
13	Reserved		
14	Reserved		
15	Reserved		
16	EMIF(DDR3)	Logic can be powered down	Software control
17	Reserved		
18	PCIe1	Logic can be powered down	Software control
19	Reserved		
20	Reserved		
21	Reserved		
22	Reserved		
23	Reserved		
24	Reserved		
25	Reserved		
26	Reserved		
27	Reserved		
28	Reserved		
29	10GbE	Logic can be powered down	Software control
30	ARM Smart Reflex	Logic can be powered down	Software control

Table 10-6. AM66K2Ex Power Domains (continued)

DOMAIN	BLOCK(S)	NOTE	POWER CONNECTION
31	ARM CorePac	Logic can be powered down	Software control

10.3.2 Clock Domains

Clock gating to each logic block is managed by the Local Power Sleep Controllers (LPSCs) of each module. For modules with a dedicated clock or multiple clocks, the LPSC communicates with the PLL controller to enable and disable that module's clock(s) at the source. For modules that share a clock with other modules, the LPSC controls the clock gating logic for each module.

Table 10-7 shows the AM5K2E0x clock domains.

Table 10-7. Clock Domains

LPSC NUMBER	MODULE(S)	NOTES
0	Shared LPSC for all peripherals other than those listed in this table	Always on
1	USB_1	
2	USB_0	Software control
3	EMIF16 and SPI	Software control
4	TSIP	Software control
5	Debug subsystem and tracers	Software control
6	Reserved	Always on
7	Packet Accelerator	Software control
8	Ethernet SGMIIs	Software control
9	Security Accelerator	Software control
10	PCIe_0	Software control
11	Reserved	
12	HyperLink	Software control
13	SmartReflex	Always on
14	MSMC RAM	Software control
15	Reserved	
16	Reserved	
17	Reserved	
18	Reserved	
19	Reserved	
20	Reserved	
21	Reserved	
22	Reserved	
23	DDR3 EMIF	Software control
24	Reserved	
25	Reserved	Reserved
26	Reserved	Reserved
27	PCIe_1	Reserved
28	Reserved	Reserved
29	Reserved	Reserved
30	Reserved	Reserved
31	Reserved	Reserved
32	Reserved	Reserved
33	Reserved	Reserved
34	Reserved	Reserved

Table 10-7. Clock Domains (continued)

LPSC NUMBER	MODULE(S)	NOTES
35	Reserved	Reserved
36	Reserved	Reserved
37	Reserved	Reserved
38	Reserved	Reserved
39	Reserved	Reserved
40	Reserved	Reserved
41	Reserved	Reserved
42	Reserved	Reserved
43	Reserved	Reserved
44	Reserved	Reserved
45	Reserved	Reserved
46	Reserved	Reserved
47	Reserved	Reserved
48	Reserved	Reserved
49	Reserved	Reserved
50	10GbE	Software control
51	ARM Smart Reflex	Software control
52	ARM CorePac	Software control
No LPSC	Bootcfg, PSC, and PLL Controller	These modules do not use LPSC

10.3.3 PSC Register Memory Map

Table 10-8 shows the PSC Register memory map.

Table 10-8. PSC Register Memory Map

OFFSET	REGISTER	DESCRIPTION
0x000	PID	Peripheral Identification Register
0x004 - 0x010	Reserved	Reserved
0x014	VCNTLID	Voltage Control Identification Register
0x018 - 0x11C	Reserved	Reserved
0x120	PTCMD	Power Domain Transition Command Register
0x124	Reserved	Reserved
0x128	PTSTAT	Power Domain Transition Status Register
0x12C - 0x1FC	Reserved	Reserved
0x200	PDSTAT0	Power Domain Status Register 0
0x204	PDSTAT1	Power Domain Status Register 1
0x208	PDSTAT2	Power Domain Status Register 2
0x20C	PDSTAT3	Power Domain Status Register 3
0x210	PDSTAT4	Power Domain Status Register 4
0x214	PDSTAT5	Power Domain Status Register 5
0x218	PDSTAT6	Power Domain Status Register 6
0x21C	PDSTAT7	Power Domain Status Register 7
0x220	PDSTAT8	Power Domain Status Register 8
0x224	PDSTAT9	Power Domain Status Register 9
0x228	PDSTAT10	Power Domain Status Register 10
0x22C	PDSTAT11	Power Domain Status Register 11
0x230	PDSTAT12	Power Domain Status Register 12
0x234	PDSTAT13	Power Domain Status Register 13

Table 10-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
0x238	PDSTAT14	Power Domain Status Register 14
0x23C	PDSTAT15	Power Domain Status Register 15
0x240	PDSTAT16	Power Domain Status Register 16
0x244	PDSTAT17	Power Domain Status Register 17
0x248	PDSTAT18	Power Domain Status Register 18
0x24C	PDSTAT19	Power Domain Status Register 19
0x250	PDSTAT20	Power Domain Status Register 20
0x254	PDSTAT21	Power Domain Status Register 21
0x258	PDSTAT22	Power Domain Status Register 22
0x25C	PDSTAT23	Power Domain Status Register 23
0x260	PDSTAT24	Power Domain Status Register 24
0x264	PDSTAT25	Power Domain Status Register 25
0x268	PDSTAT26	Power Domain Status Register 26
0x26C	PDSTAT27	Power Domain Status Register 27
0x270	PDSTAT28	Power Domain Status Register 28
0x274	PDSTAT29	Power Domain Status Register 29
0x278	PDSTAT30	Power Domain Status Register 30
0x27C	PDSTAT31	Power Domain Status Register 31
0x27C - 0x2FC	Reserved	Reserved
0x300	PDCTL0	Power Domain Control Register 0
0x304	PDCTL1	Power Domain Control Register 1
0x308	PDCTL2	Power Domain Control Register 2
0x30C	PDCTL3	Power Domain Control Register 3
0x310	PDCTL4	Power Domain Control Register 4
0x314	PDCTL5	Power Domain Control Register 5
0x318	PDCTL6	Power Domain Control Register 6
0x31C	PDCTL7	Power Domain Control Register 7
0x320	PDCTL8	Power Domain Control Register 8
0x324	PDCTL9	Power Domain Control Register 9
0x328	PDCTL10	Power Domain Control Register 10
0x32C	PDCTL11	Power Domain Control Register 11
0x330	PDCTL12	Power Domain Control Register 12
0x334	PDCTL13	Power Domain Control Register 13
0x338	PDCTL14	Power Domain Control Register 14
0x33C	PDCTL15	Power Domain Control Register 15
0x340	PDCTL16	Power Domain Control Register 16
0x344	PDCTL17	Power Domain Control Register 17
0x348	PDCTL18	Power Domain Control Register 18
0x34C	PDCTL19	Power Domain Control Register 19
0x350	PDCTL20	Power Domain Control Register 20
0x354	PDCTL21	Power Domain Control Register 21
0x358	PDCTL22	Power Domain Control Register 22
0x35c	PDCTL23	Power Domain Control Register 23
0x360	PDCTL24	Power Domain Control Register 24
0x364	PDCTL25	Power Domain Control Register 25
0x368	PDCTL26	Power Domain Control Register 26
0x36C	PDCTL27	Power Domain Control Register 27

Table 10-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
0x370	PDCTL28	Power Domain Control Register 28
0x374	PDCTL29	Power Domain Control Register 29
0x378	PDCTL30	Power Domain Control Register 30
0x37C	PDCTL31	Power Domain Control Register 31
0x380 - 0x7FC	Reserved	Reserved
0x800	MDSTAT0	Module Status Register 0 (never gated)
0x804	MDSTAT1	Module Status Register 1
0x808	MDSTAT2	Module Status Register 2
0x80C	MDSTAT3	Module Status Register 3
0x810	MDSTAT4	Module Status Register 4
0x814	MDSTAT5	Module Status Register 5
0x818	MDSTAT6	Module Status Register 6
0x81C	MDSTAT7	Module Status Register 7
0x820	MDSTAT8	Module Status Register 8
0x824	MDSTAT9	Module Status Register 9
0x828	MDSTAT10	Module Status Register 10
0x82C	MDSTAT11	Module Status Register 11
0x830	MDSTAT12	Module Status Register 12
0x834	MDSTAT13	Module Status Register 13
0x838	MDSTAT14	Module Status Register 14
0x83C	MDSTAT15	Module Status Register 15
0x840	MDSTAT16	Module Status Register 16
0x844	MDSTAT17	Module Status Register 17
0x848	MDSTAT18	Module Status Register 18
0x84C	MDSTAT19	Module Status Register 19
0x850	MDSTAT20	Module Status Register 20
0x854	MDSTAT21	Module Status Register 21
0x858	MDSTAT22	Module Status Register 22
0x85C	MDSTAT23	Module Status Register 23
0x860	MDSTAT24	Module Status Register 24
0x864	MDSTAT25	Module Status Register 25
0x868	MDSTAT26	Module Status Register 26
0x86C	MDSTAT27	Module Status Register 27
0x870	MDSTAT28	Module Status Register 28
0x874	MDSTAT29	Module Status Register 29
0x878	MDSTAT30	Module Status Register 30
0x87C	MDSTAT31	Module Status Register31
0x880	MDSTAT32	Module Status Register 32
0x884	MDSTAT33	Module Status Register 33
0x888	MDSTAT34	Module Status Register 34
0x88C	MDSTAT35	Module Status Register 35
0x890	MDSTAT36	Module Status Register 36
0x894	MDSTAT37	Module Status Register 37
0x898	MDSTAT38	Module Status Register 38
0x89C	MDSTAT39	Module Status Register 39
0x8A0	MDSTAT40	Module Status Register 40
0x8A4	MDSTAT41	Module Status Register 41

Table 10-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
0x8A8	MDSTAT42	Module Status Register 42
0x8AC	MDSTAT43	Module Status Register 43
0x8B0	MDSTAT44	Module Status Register 44
0x8B4	MDSTAT45	Module Status Register 45
0x8B8	MDSTAT46	Module Status Register 46
0x8BC	MDSTAT47	Module Status Register 47
0x8C0	MDSTAT48	Module Status Register 48
0x8C4	MDSTAT49	Module Status Register 49
0x8C8	MDSTAT50	Module Status Register 50
0x8CC	MDSTAT51	Module Status Register 51
0x8D0	MDSTAT52	Module Status Register 52
0x8D4 - 0x9FC	Reserved	Reserved
0xA00	MDCTL0	Module Control Register 0 (never gated)
0xA04	MDCTL1	Module Control Register 1
0xA08	MDCTL2	Module Control Register 2
0xA0C	MDCTL3	Module Control Register 3
0xA10	MDCTL4	Module Control Register 4
0xA14	MDCTL5	Module Control Register 5
0xA18	MDCTL6	Module Control Register 6
0xA1C	MDCTL7	Module Control Register 7
0xA20	MDCTL8	Module Control Register 8
0xA24	MDCTL9	Module Control Register 9
0xA28	MDCTL10	Module Control Register 10
0xA2C	MDCTL11	Module Control Register 11
0xA30	MDCTL12	Module Control Register 12
0xA34	MDCTL13	Module Control Register 13
0xA38	MDCTL14	Module Control Register 14
0xA3C	MDCTL15	Module Control Register 15
0xA40	MDCTL16	Module Control Register 16
0xA44	MDCTL17	Module Control Register 17
0xA48	MDCTL18	Module Control Register 18
0xA4C	MDCTL19	Module Control Register 19
0xA50	MDCTL20	Module Control Register 20
0xA54	MDCTL21	Module Control Register 21
0xA58	MDCTL22	Module Control Register 22
0xA5C	MDCTL23	Module Control Register 23
0xA60	MDCTL24	Module Control Register 24
0xA64	MDCTL25	Module Control Register 25
0xA68	MDCTL26	Module Control Register 26
0xA6C	MDCTL27	Module Control Register 27
0xA70	MDCTL28	Module Control Register 28
0xA74	MDCTL29	Module Control Register 29
0xA78	MDCTL30	Module Control Register 30
0xA7C	MDCTL31	Module Control Register31
0xA80	MDCTL32	Module Control Register 32
0xA84	MDCTL33	Module Control Register 33
0xA88	MDCTL34	Module Control Register 34

Table 10-8. PSC Register Memory Map (continued)

OFFSET	REGISTER	DESCRIPTION
0xA8C	MDCTL35	Module Control Register 35
0xA90	MDCTL36	Module Control Register 36
0xA94	MDCTL37	Module Control Register 37
0xA98	MDCTL38	Module Control Register 38
0xA9C	MDCTL39	Module Control Register 39
0xAA0	MDCTL40	Module Control Register 40
0xAA4	MDCTL41	Module Control Register 41
0xAA8	MDCTL42	Module Control Register 42
0xAAC	MDCTL43	Module Control Register 43
0xAB0	MDCTL44	Module Control Register 44
0xAB4	MDCTL45	Module Control Register 45
0xAB8	MDCTL46	Module Control Register 46
0xABC	MDCTL47	Module Control Register 47
0xAC0	MDCTL48	Module Control Register 48
0xAC4	MDCTL49	Module Control Register 49
0xAC8	MDCTL50	Module Control Register 50
0xACC	MDCTL51	Module Control Register 51
0xAD0	MDCTL52	Module Control Register 52
0xAD4 - 0xFFC	Reserved	Reserved

10.4 Reset Controller

The reset controller detects the different type of resets supported on the AM5K2E0x device and manages the distribution of those resets throughout the device. The device has the following types of resets:

- Power-on reset
- Hard reset
- Soft reset
- Local reset

Table 10-9 explains further the types of reset, the reset initiator, and the effects of each reset on the device. For more information on the effects of each reset on the PLL controllers and their clocks, see Section 10.4.8.

Table 10-9. Reset Types

TYPE	INITIATOR	EFFECT(S)
Power-on reset	$\overline{\text{POR}}$ pin $\overline{\text{RESETFULL}}$ pin	Resets the entire chip including the test and emulation logic. The device configuration pins are latched only during power-on reset.
Hard reset	$\overline{\text{RESET}}$ pin PLLCTL Register (RSCTRL)(1) Watchdog timers Emulation	Hard reset resets everything except for test, emulation logic, and reset isolation modules. This reset is different from power-on reset in that the PLL Controller assumes power and clocks are stable when a hard reset is asserted. The device configurations pins are not relatched. Emulation-initiated reset is always a hard reset. By default, these initiators are configured as hard reset, but can be configured (except emulation) as a soft reset in the RSCFG Register of the PLL Controller. Contents of the DDR3 SDRAM memory can be retained during a hard reset if the SDRAM is placed in self-refresh mode.

(1) All masters in the device have access to the PLL Control Registers.

Table 10-9. Reset Types (continued)

TYPE	INITIATOR	EFFECT(S)
Soft reset	$\overline{\text{RESET}}$ pin PLLCTL Register (RSCTRL) Watchdog timers	Soft reset behaves like hard reset except that PCIe MMRs (memory-mapped registers) and DDR3 EMIF MMRs contents are retained. By default, these initiators are configured as hard reset, but can be configured as soft reset in the RSCFG Register of the PLL Controller. Contents of the DDR3 SDRAM memory can be retained during a soft reset if the SDRAM is placed in self-refresh mode.
Local reset	$\overline{\text{LRESET}}$ pin Watchdog timer timeout LPSC MMRs	Resets the C66x CorePac, without disturbing clock alignment or memory contents. The device configuration pins are not relatched.

10.4.1 Power-on Reset

Power-on reset is used to reset the entire device, including the test and emulation logic.

Power-on reset is initiated by the following:

1. $\overline{\text{POR}}$ pin
2. $\overline{\text{RESETFULL}}$ pin

During power-up, the $\overline{\text{POR}}$ pin must be asserted (driven low) until the power supplies have reached their normal operating conditions. Also a $\overline{\text{RESETFULL}}$ pin is provided to allow reset of the entire device, including the reset-isolated logic, when the device is already powered up. For this reason, the $\overline{\text{RESETFULL}}$ pin, unlike $\overline{\text{POR}}$, should be driven by the on-board host control other than the power good circuitry. For power-on reset, the Core PLL Controller comes up in bypass mode and the PLL is not enabled. Other resets do not affect the state of the PLL or the dividers in the PLL Controller.

The following sequence must be followed during a power-on reset:

1. Wait for all power supplies to reach normal operating conditions while keeping the $\overline{\text{POR}}$ and $\overline{\text{RESETFULL}}$ pins asserted (driven low). While $\overline{\text{POR}}$ is asserted, all pins except $\overline{\text{RESETSTAT}}$ will be set to high-impedance. After the $\overline{\text{POR}}$ pin is deasserted (driven high), all Z group pins, low group pins, and high group pins are set to their reset state and remain in their reset state until otherwise configured by their respective peripheral. All peripherals that are power-managed are disabled after a power-on reset and must be enabled through the Device State Control Registers (for more details, see Section 8.2.3).
2. Clocks are reset, and they are propagated throughout the chip to reset any logic that was using reset synchronously. All logic is now reset and $\overline{\text{RESETSTAT}}$ is driven low, indicating that the device is in reset.
3. $\overline{\text{POR}}$ and $\overline{\text{RESETFULL}}$ must be held active until all supplies on the board are stable, and then for at least an additional period of time (as specified in Section 10.2.1) for the chip-level PLLs to lock.
4. The $\overline{\text{POR}}$ pin can now be de-asserted.
5. After the appropriate delay, the $\overline{\text{RESETFULL}}$ pin can now be de-asserted. Reset-sampled pin values are latched at this point. Then, all chip-level PLLs are taken out of reset, locking sequences begin, and all power-on device initialization processes begin.
6. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is de-asserted (driven high). By this time, the DDR3 PLL has completed its locking sequences and are supplying a valid clock. The system clocks of the PLL controllers are allowed to finish their current cycles and then are paused for 10 cycles of their respective system reference clocks. After the pause, the system clocks are restarted at their default divide-by settings.
7. The device is now out of reset and code execution begins as dictated by the selected boot mode.

NOTE

To most of the device, reset is de-asserted only when the $\overline{\text{POR}}$ and $\overline{\text{RESET}}$ pins are both de-asserted (driven high). Therefore, in the sequence described above, if the $\overline{\text{RESET}}$ pin is held low past the low period of the $\overline{\text{POR}}$ pin, most of the device will remain in reset. The $\overline{\text{RESET}}$ pin should not be tied to the $\overline{\text{POR}}$ pin.

10.4.2 Hard Reset

A hard reset will reset everything on the device except the PLLs, test logic, emulation logic, and reset-isolated modules. $\overline{\text{POR}}$ should also remain de-asserted during this time.

Hard reset is initiated by the following:

- $\overline{\text{RESET}}$ pin
- RSCTRL Register in the PLL Controller
- Watchdog timer
- Emulation

By default, all the initiators listed above are configured to generate a hard reset. Except for emulation, all of the other three initiators can be configured in the RSCFG Register in the PLL Controller to generate soft resets.

The following sequence must be followed during a hard reset:

1. The $\overline{\text{RESET}}$ pin is asserted (driven low) for a minimum of 24 CLKIN1 cycles. During this time, the $\overline{\text{RESET}}$ signal propagates to all modules (except those specifically mentioned above). To prevent off-chip contention during the warm reset, all I/O must be Hi-Z for modules affected by $\overline{\text{RESET}}$.
2. Once all logic is reset, $\overline{\text{RESETSTAT}}$ is asserted (driven low) to denote that the device is in reset.
3. The $\overline{\text{RESET}}$ pin can now be released. A minimal device initialization begins to occur. Note that configuration pins are not re-latched and clocking is unaffected within the device.
4. After device initialization is complete, the $\overline{\text{RESETSTAT}}$ pin is de-asserted (driven high).

NOTE

The $\overline{\text{POR}}$ pin should be held inactive (high) throughout the warm reset sequence. Otherwise, if $\overline{\text{POR}}$ is activated (brought low), the minimum $\overline{\text{POR}}$ pulse width must be met. The $\overline{\text{RESET}}$ pin should not be tied to the $\overline{\text{POR}}$ pin.

10.4.3 Soft Reset

A soft reset behaves like a hard reset except that the EMIF16 MMRs, DDR3 EMIF MMRs, PCIe MMRs sticky bits, and external memory content are retained. $\overline{\text{POR}}$ should also remain de-asserted during this time.

Soft reset is initiated by the following:

- $\overline{\text{RESET}}$ pin
- RSCTRL Register in the PLL Controller
- Watchdog timer

In the case of a soft reset, the clock logic and the power control logic of the peripherals are not affected and, therefore, the enabled/disabled state of the peripherals is not affected. On a soft reset, the DDR3 memory controller registers are ***not*** reset. If the user places the DDR3 SDRAM in self-refresh mode before invoking the soft reset, the DDR3 SDRAM memory content is retained.

During a soft reset, the following occurs:

1. The $\overline{\text{RESETSTAT}}$ pin goes low to indicate an internal reset is being generated. The reset propagates through the system. Internal system clocks are not affected. PLLs remain locked.

2. After device initialization is complete, the RESETSTAT pin is deasserted (driven high). In addition, the PLL Controller pauses system clocks for approximately 8 cycles. At this point:
   – The peripherals remain in the state they were in before the soft reset.
   – The states of the Boot Mode configuration pins are preserved as controlled by the DEVSTAT Register.
   – The DDR3 MMRs and PCIe MMRs retain their previous values. Only the DDR3 memory controller and PCIe state machines are reset by the soft reset.
   – The PLL Controller remains in the mode it was in prior to the soft reset.
   – System clocks are unaffected.

The boot sequence is started after the system clocks are restarted. Because the Boot Mode configuration pins are not latched with a soft reset, the previous values (as shown in the DEVSTAT Register), are used to select the boot mode.

10.4.4 Local Reset

The local reset can be used to reset a particular C66x CorePac without resetting any other device components.

Local reset is initiated by the following:

- LRESET pin
- Watchdog timer should cause one of the below and RSTCFG registers in the PLL Controller. (See Section 10.5.2.8 and Section 6.3.2.)
  – Local reset
  – NMI
  – NMI followed by a time delay and then a local reset for the C66x CorePac selected
  – Hard reset by requesting reset via the PLL Controller
- LPSC MMRs (memory-mapped registers)

For more details see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

10.4.5 ARM CorePac Reset

The ARM CorePac uses a combination of power-on-reset and module-reset to reset its components, such as the Cortex-A15 processor, memory subsystem, debug logic, etc. The ARM CorePac incorporates the PSC to generate resets for its internal modules. Details of reset generation and distribution inside the ARM CorePac can be found in the *KeyStone II Architecture ARM CorePac User's Guide* (SPRUHJ4).

10.4.6 Reset Priority

If any of the above reset sources occur simultaneously, the PLL Controller processes only the highest priority reset request. The reset request priorities are as follows (high to low):

- Power-on reset
- Hard/soft reset

10.4.7 Reset Controller Register

The reset controller registers are part of the PLL Controller MMRs. All AM5K2E0x device-specific MMRs are covered in Section 10.5.2. For more details on these registers and how to program them, see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

10.4.8 Reset Electrical Data/Timing

Table 10-10. Reset Timing Requirements(1)

(see Figure 10-4 and Figure 10-5)

NO.			MIN	MAX	UNIT
	$\overline{\text{RESETFULL}}$ Pin Reset				
1	tw($\overline{\text{RESETFULL}}$)	Pulse width - pulse width $\overline{\text{RESETFULL}}$ low	500C		ns
	Soft/Hard-Reset				
2	tw($\overline{\text{RESET}}$)	Pulse width - pulse width $\overline{\text{RESET}}$ low	500C		ns

(1) C = 1/SYSCLK1 clock frequency in ns

Table 10-11. Reset Switching Characteristics(1)

(see Figure 10-4 and Figure 10-5)

NO.	PARAMETER		MIN	MAX	UNIT
	$\overline{\text{RESETFULL}}$ Pin Reset				
3	td($\overline{\text{RESETFULL}}$LH-$\overline{\text{RESETSTAT}}$H)	Delay time - $\overline{\text{RESETSTAT}}$ high after $\overline{\text{RESETFULL}}$ high		50000C	ns
	Soft/Hard Reset				
4	td($\overline{\text{RESET}}$H-$\overline{\text{RESETSTAT}}$H)	Delay time - $\overline{\text{RESETSTAT}}$ high after $\overline{\text{RESET}}$ high		50000C	ns

(1) C = 1/SYSCLK1 clock frequency in ns

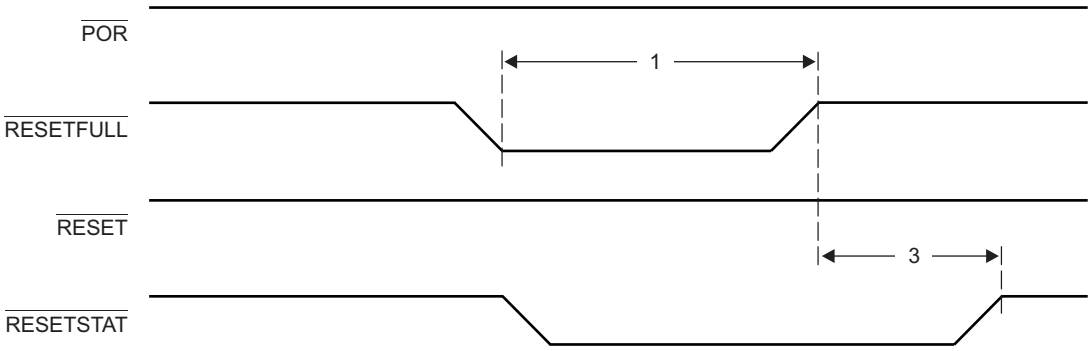


Figure 10-4. $\overline{\text{RESETFULL}}$ Reset Timing

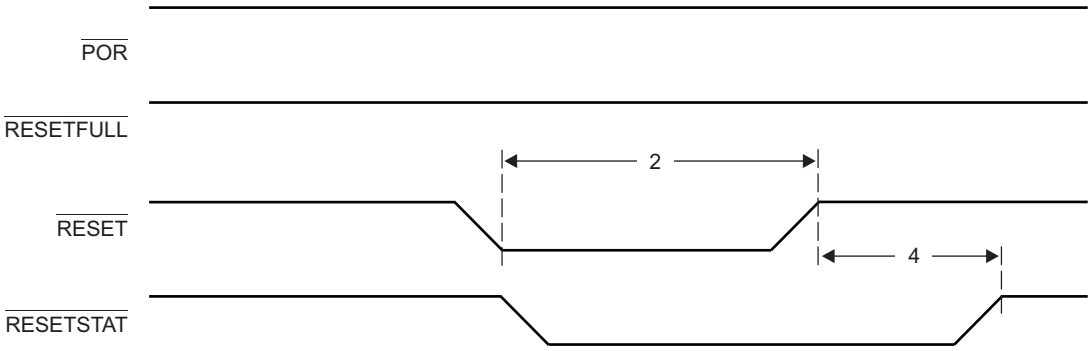


Figure 10-5. Soft/Hard Reset Timing

Table 10-12. Boot Configuration Timing Requirements(1)

(see Figure 10-6)

NO.			MIN	MAX	UNIT
1	tsu(GPIOn-$\overline{\text{RESETFULL}}$)	Setup time - GPIO valid before $\overline{\text{RESETFULL}}$ asserted	12C		ns
2	th($\overline{\text{RESETFULL}}$-GPIOn)	Hold time - GPIO valid after $\overline{\text{RESETFULL}}$ asserted	12C		ns

(1) C = 1/SYSCLK1 clock frequency in ns.

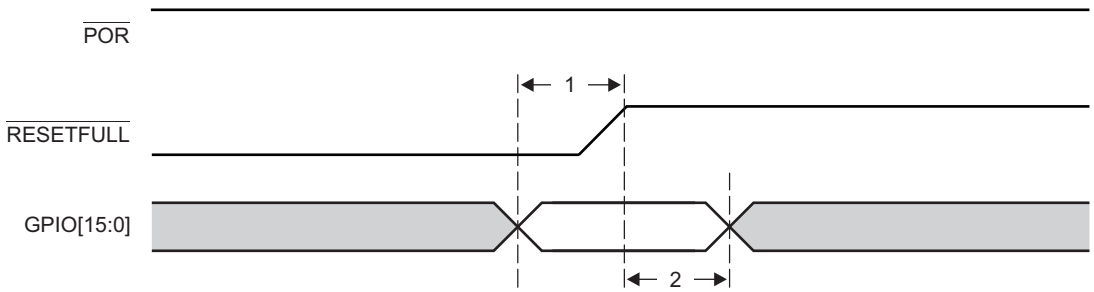


Figure 10-6. Boot Configuration Timing

10.5 Core PLL (Main PLL), DDR3 PLL, NETCP PLL and the PLL Controllers

This section provides a description of the Core PLL, DDR3 PLL, NETCP PLL, and the PLL Controller. For details on the operation of the PLL Controller module, see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

The Core PLL is controlled by the standard PLL Controller. The PLL Controller manages the clock ratios, alignment, and gating for the system clocks to the device. By default, the device powers up with the Core PLL bypassed. Figure 10-7 shows a block diagram of the Core PLL and the PLL Controller.

The DDR3 PLL and NETCP PLL are used to provide dedicated clock to the DDR3 and NETCP respectively. These chip level PLLs support a wide range of multiplier and divider values, which can be programmed through the chip level registers located in the Device Control Register block. The Boot ROM will program the multiplier values for Core PLL and NETCP PLL based on boot mode. (See Section 8 for more details.)

The DDR3 PLL is used to supply clocks to DDR3 EMIF logic. This PLL can also be used without programming the PLL Controller. Instead, they can be controlled using the chip-level registers (DDR3PLLCTL0, DDR3PLLCTL1) located in the Device Control Register block. To write to these registers, software must go through an unlocking sequence using the KICK0/KICK1 registers.

The multiplier values for all chip-level PLLs can be reprogrammed later based on the input parameter table. This feature provides flexibility in that these PLLs may be able to reuse other clock sources instead of having its own clock source.

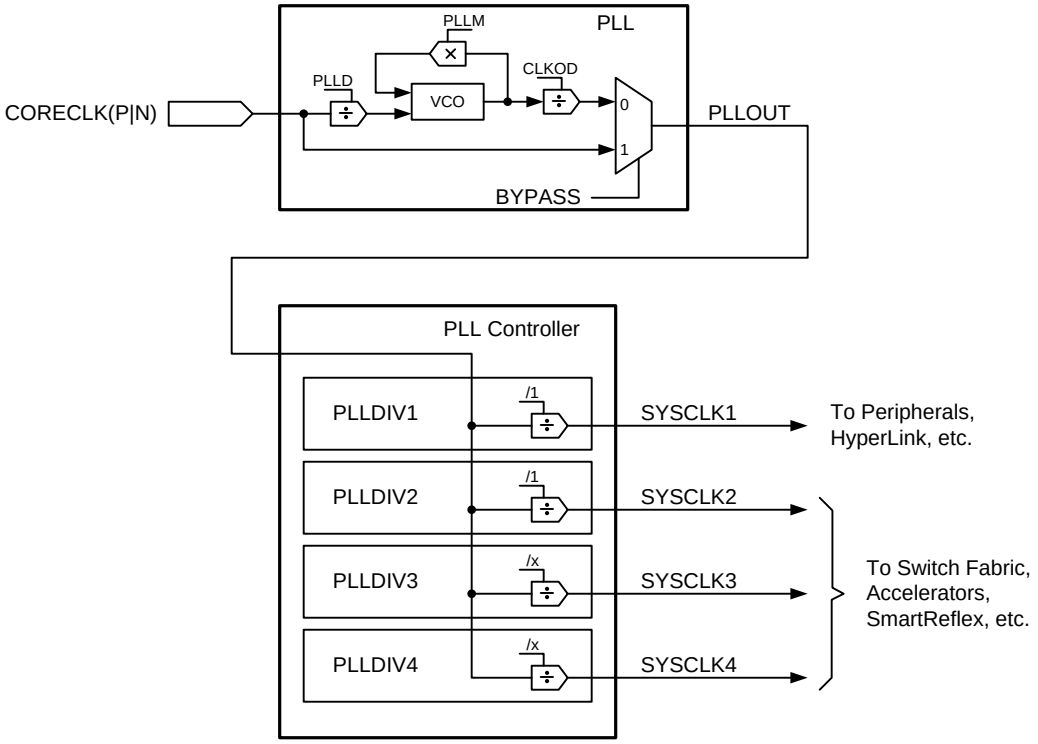


Figure 10-7. Core PLL and PLL Controller

Note that the Core PLL Controller registers can be accessed by any master in the device. The PLLM[5:0] bits of the multiplier are controlled by the PLLM Register inside the PLL Controller and the PLLM[12:6] bits are controlled by the chip-level COREPLLCTL0 Register. The output divide and bypass logic of the PLL are controlled by fields in the SECCTL Register in the PLL Controller. Only PLLDIV3, and PLLDIV4 are programmable on the device. See the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2) for more details on how to program the PLL controller.

The multiplication and division ratios within the PLL and the post-division for each of the chip-level clocks are determined by a combination of this PLL and the Core PLL Controller. The Core PLL Controller also controls reset propagation through the chip, clock alignment, and test points. The Core PLL Controller monitors the PLL status and provides an output signal indicating when the PLL is locked.

Core PLL power is supplied externally via the Core PLL power-supply pin (AVDDA1). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0) for detailed recommendations. For the best performance, TI recommends that all the PLL external components be on a single side of the board without jumpers, switches, or components other than those shown. For reduced PLL jitter, maximize the spacing between switching signal traces and the PLL external components (C1, C2, and the EMI Filter).

The minimum CORECLK rise and fall times should also be observed. For the input clock timing requirements, see Section 10.5.4.

It should be assumed that any registers not included in these sections are not supported by the device. Furthermore, only the bits within the registers described here are supported. Avoid writing to any reserved memory location or changing the value of reserved bits.

The PLL Controller module as described in the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2) includes a superset of features, some of which are not supported on the device. The following sections describe the registers that are supported.

10.5.1 Core PLL Controller Device-Specific Information

10.5.1.1 Internal Clocks and Maximum Operating Frequencies

The Core PLL, used to drive the SoC, the switch fabric, and a majority of the peripheral clocks (all but the ARM CorePacs, DDR3, and the NETCP modules) requires a PLL Controller to manage the various clock divisions, gating, and synchronization. PLLM[5:0] input of the Core PLL is controlled by the PLL controller PLLM register.

The Core PLL Controller has four SYSCLK outputs that are listed below along with the clock descriptions. Each SYSCLK has a corresponding divider that divides down the output clock of the PLL. Note that dividers are not programmable unless explicitly mentioned in the description below.

- **SYSCLK1:** Using local dividers, SYSCLK1 is used to derive clocks required for the majority of peripherals that do not need reset isolation.

  The system peripherals and modules driven by SYSCLK1 are as follows; however, not all peripherals are supported in every part. See the Features chapter for the complete list of peripherals supported in your part.

  EMIF16, USB 3.0, XFI, HyperLink, PCIe, SGMII, GPIO, Timer64, I^2C, SPI, TSIP, TeraNet, UART, ROM, CIC, Security Manager, BootCFG, PSC, Queue Manager, Semaphore, MPUs, EDMA, MSMC, DDR3, EMIF.
- **SYSCLK2:**Full-rate, reset-isolated clock used to generate various other clocks required by peripherals that need reset isolation: e.g., SmartReflex.
- **SYSCLK3:** The default rate for this clock is 1/3. This clock is programmable from /1 to /32, where this clock does not violate the maximum of 350 MHz. SYSCLK3 can be turned off by software.
- **SYSCLK4:** 1/z-rate clock for the system trace module only. The default rate for this clock is 1/5. This clock is configurable: the maximum configurable clock is 210 MHz and the minimum configuration clock is 32 MHz. SYSCLK4 can be turned off by software.

Only SYSCLK3 and SYSCLK4 are programmable.

10.5.1.2 Local Clock Dividers

The clock signals from the Core PLL Controller are routed to various modules and peripherals on the device. Some modules and peripherals have one or more internal clock dividers. Other modules and peripherals have no internal clock dividers, but are grouped together and receive clock signals from a shared local clock divider. Internal and shared local clock dividers have fixed division ratios. See table Table 10-13.

Table 10-13. Core PLL Controller Module Clock Domains Internal and Shared Local Clock Dividers

CLOCK	MODULE	INTERNAL CLOCK DIVIDER(S)	SHARED LOCAL CLOCK DIVIDER
SYSCLK1 Internal Clock Dividers			
SYSCLK1	ARM CorePac	/1, /3, /3, /6, /6	--
	Chip Interrupt Controllers (CICx)	/6	--
	DDR3 Memory Controller A (also receives clocks from the DDR3_PLL)	/2	--
	EMIF16	/6	--
	HyperLink	/2, /3, /6	--
	MultiCore Shared Memory Controller (MSMC)	/1	--
	PCI express (PCIe)	/2, /3, /4, /6	--
	ROM	/6	--
	Serial Gigabit Media Independent Interface (SGMII)	/2, /3, /6, /8	--
	Universal Asynchronous Receiver/Transmitter (UART)	/6	--
	Universal Serial Bus 3.0 (USB 3.0)	/3, /6	--
SYSCLK1 Shared Local Clock Dividers			
SYSCLK1	Power/Sleep Controller (PSC)	--	/12, /24
	EDMA	--	/3
	Memory Protection Units (MPUx)		
	Semaphore		
	TeraNet (SYSCLK1/3 domain)		
SYSCLK1	Boot Config	--	/6
	General-Purpose Input/Output (GPIO)		
	I^2C		
	Security Manager		
	Telecom Serial Interface Port (TSIP)		
	Serial Peripheral Interconnect (SPI)		
	TeraNet (CPU /6 domain)		
	Timers		

10.5.1.3 Module Clock Input

Table 10-7 lists various clock domains in the device and their distribution in each peripheral. The table also shows the distributed clock division in modules and their mapping with source clocks of the device PLLs.

10.5.1.4 Core PLL Controller Operating Modes

The Core PLL Controller has two modes of operation: bypass mode and PLL mode. The mode of operation is determined by the BYPASS bit of the PLL Secondary Control Register (SECCTL).

- In bypass mode, PLL input is fed directly out as SYSCLK1.
- In PLL mode, SYSCLK1 is generated from the PLL output using the values set in the PLLM and PLLD fields in the COREPLLCTL0 Register.

External hosts must avoid access attempts to the SoC while the frequency of its internal clocks is changing. User software must implement a mechanism that causes the SoC to notify the host when the PLL configuration has completed.

10.5.1.5 Core PLL Stabilization, Lock, and Reset Times

The PLL stabilization time is the amount of time that must be allotted for the internal PLL regulators to become stable after device power-up. The device should not be taken out of reset until this stabilization time has elapsed.

The PLL reset time is the amount of wait time needed when resetting the PLL (writing PLLRST = 1), in order for the PLL to properly reset, before bringing the PLL out of reset (writing PLLRST = 0). For the Core PLL reset time value, see Table 10-14.

The PLL lock time is the amount of time needed from when the PLL is taken out of reset to when the PLL Controller can be switched to PLL mode. The Core PLL lock time is given in Table 10-14.

Table 10-14. Core PLL Stabilization, Lock, and Reset Times

PARAMETER	MIN	TYP	MAX	UNIT
PLL stabilization time	100			µs
PLL lock time			2000 × C(1)	
PLL reset time	1000			ns

(1) C = SYSCLK1(N|P) cycle time in ns.

10.5.2 PLL Controller Memory Map

The memory map of the Core PLL Controller is shown in Table 10-15. AM5K2Exx-specific Core PLL Controller Register definitions can be found in the sections following Table 10-15. For other registers in the table, see the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2).

It is recommended to use read-modify-write sequence to make any changes to the valid bits in the Core PLL Controller registers.

Note that only registers documented here are accessible on the AM5K2Exx. Other addresses in the Core PLL Controller memory map including the Reserved registers must not be modified. Furthermore, only the bits within the registers described here are supported.

Table 10-15. PLL Controller Registers (Including Reset Controller)

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
00 0231 0000 - 00 0231 00E3	-	Reserved
00 0231 00E4	RSTYPE	Reset Type Status Register (Reset Core PLL Controller)
00 0231 00E8	RSTCTRL	Software Reset Control Register (Reset Core PLL Controller)
00 0231 00EC	RSTCFG	Reset Configuration Register (Reset Core PLL Controller)
00 0231 00F0	RSISO	Reset Isolation Register (Reset Core PLL Controller)
00 0231 00F0 - 00 0231 00FF	-	Reserved
00 0231 0100	PLLCTL	PLL Control Register
00 0231 0104	-	Reserved
00 0231 0108	SECCTL	PLL Secondary Control Register
00 0231 010C	-	Reserved
00 0231 0110	PLLM	PLL Multiplier Control Register
00 0231 0114	-	Reserved
00 0231 0118	PLLDIV1	PLL Controller Divider 1Register
00 0231 011C	PLLDIV2	PLL Controller Divider 2 Register
00 0231 0120	PLLDIV3	PLL Controller Divider 3Register
00 0231 0124	-	Reserved
00 0231 0128	-	Reserved
00 0231 012C - 00 0231 0134	-	Reserved
00 0231 0138	PLLCMD	PLL Controller Command Register
00 0231 013C	PLLSTAT	PLL Controller Status Register
00 0231 0140	ALNCTL	PLL Controller Clock Align Control Register
00 0231 0144	DCHANGE	PLLDIV Ratio Change Status Register
00 0231 0148	CKEN	Reserved
00 0231 014C	CKSTAT	Reserved

Table 10-15. PLL Controller Registers (Including Reset Controller) (continued)

HEX ADDRESS RANGE	ACRONYM	REGISTER NAME
00 0231 0150	SYSTAT	SYSCLK Status Register
00 0231 0154 - 00 0231 015C	-	Reserved
00 0231 0160	PLLDIV4	PLL Controller Divider 4Register
00 0231 0164	PLLDIV5	Reserved
00 0231 0168	PLLDIV6	Reserved
00 0231 016C	PLLDIV7	Reserved
00 0231 0170	PLLDIV8	Reserved
00 0231 0174 - 00 0231 0193	PLLDIV9 - PLLDIV16	Reserved
00 0231 0194 - 00 0231 01FF	-	Reserved

10.5.2.1 PLL Secondary Control Register (SECCTL)

The PLL Secondary Control Register contains extra fields to control the Core PLL and is shown in Figure 10-8 and described in Table 10-16.

Figure 10-8. PLL Secondary Control Register (SECCTL)

31	24	23	22	19	18	0
Reserved		BYPASS	OUTPUT DIVIDE		Reserved	
R-0000 0000		RW-1	RW-0001		RW-001 0000 0000 0000 0000	

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-16. PLL Secondary Control Register Field Descriptions

Bit	Field	Description
31-24	Reserved	Reserved
23	BYPASS	Core PLL bypass enable • 0 - Core PLL bypass disabled • 1 - Core PLL bypass enabled
22-19	OUTPUT DIVIDE	Output divider ratio bits • 0h - ÷1. Divide frequency by 1 • 1h - ÷2. Divide frequency by 2 • 2h - invalid entry • 3h - ÷4. Divide frequency by 4 • 4h - invalid entry • 5h - ÷6. Divide frequency by 6 • 6h - invalid entry • 7h - ÷8. Divide frequency by 8 • 8h - invalid entry • 9h - ÷10. Divide frequency by 10 • Ah - invalid entry • Bh - ÷12. Divide frequency by 12 • Ch - invalid entry • Dh - ÷14. Divide frequency by 14 • Eh - invalid entry • Fh - ÷16. Divide frequency by 16
18-0	Reserved	Reserved

10.5.2.2 PLL Controller Divider Register (PLLDIV3, and PLLDIV4)

The PLL Controller Divider Registers (PLLDIV3 and PLLDIV4) are shown in Figure 10-9 and described in Table 10-17. The default values of the RATIO field on a reset for PLLDIV3, and PLLDIV4 are different as mentioned in the footnote of Figure 10-9.

Figure 10-9. PLL Controller Divider Register (PLLDIVn)

31	16	15	14	8	7	0
Reserved		D*n*(1)EN	Reserved		RATIO	
R-0		R/W-1	R-0		R/W-n(2)	

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset

(1) D3EN for PLLDIV3; D4EN for PLLDIV4
(2) n=02h for PLLDIV3; n=03h for PLLDIV4

Table 10-17. PLL Controller Divider Register Field Descriptions

Bit	Field	Description
31-16	Reserved	Reserved
15	D*n*EN	Divider D*n* enable bit (See footnote of Figure 10-9) • 0 = Divider *n* is disabled • 1 = No clock output. Divider *n* is enabled.
14-8	Reserved	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
7-0	RATIO	Divider ratio bits (See footnote of Figure 10-9) • 0h = ÷1. Divide frequency by 1 • 1h = ÷2. Divide frequency by 2 • 2h = ÷3. Divide frequency by 3 • 3h = ÷4. Divide frequency by 4 • 4h - 4Fh = ÷5 to ÷80. Divide frequency range: divide frequency by 5 to divide frequency by 80.

10.5.2.3 PLL Controller Clock Align Control Register (ALNCTL)

The PLL Controller Clock Align Control Register (ALNCTL) is shown in Figure 10-10 and described in Table 10-18.

Figure 10-10. PLL Controller Clock Align Control Register (ALNCTL)

31	5	4	3	2	0
Reserved		ALN4	ALN3	Reserved	
R-0		R/W-1	R/W-1	R-0	

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset, for reset value

Table 10-18. PLL Controller Clock Align Control Register Field Descriptions

Bit	Field	Description
31-5 2-0	Reserved	Reserved. This location is always read as 0. A value written to this field has no effect.
4 3	ALN4 ALN3	SYSCLK*n* alignment. Do not change the default values of these fields. • 0 = Do not align SYSCLK*n* to other SYSCLKs during GO operation. If SYS*n* in DCHANGE is set, SYSCLK*n* switches to the new ratio immediately after the GOSET bit in PLLCMD is set. • 1 = Align SYSCLK*n* to other SYSCLKs selected in ALNCTL when the GOSET bit in PLLCMD is set and SYS*n* in DCHANGE is 1. The SYSCLK*n* rate is set to the ratio programmed in the RATIO bit in PLLDIV*n*.

10.5.2.4 PLLDIV Divider Ratio Change Status Register (DCHANGE)

Whenever a different ratio is written to the PLLDIV*n* registers, the PLL CTL flags the change in the DCHANGE Status Register. During the GO operation, the PLL controller changes only the divide ratio of the SYSCLKs with the bit set in DCHANGE. Note that the ALNCTL Register determines if that clock also needs to be aligned to other clocks. The PLLDIV Divider Ratio Change Status Register is shown in Figure 10-11 and described in Table 10-19.

Figure 10-11. PLLDIV Divider Ratio Change Status Register (DCHANGE)

31	5	4	3	2	0
Reserved		SYS4	SYS3	Reserved	
R-0		R/W-1	R/W-1	R-0	

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset, for reset value

Table 10-19. PLLDIV Divider Ratio Change Status Register Field Descriptions

Bit	Field	Description
31-5 2-0	Reserved	Reserved. This bit location is always read as 0. A value written to this field has no effect.
4 3	SYS4 SYS3	Identifies when the SYSCLK*n* divide ratio has been modified. • 0 = SYSCLK*n* ratio has not been modified. When GOSET is set, SYSCLK*n* will not be affected. • 1 = SYSCLK*n* ratio has been modified. When GOSET is set, SYSCLK*n* will change to the new ratio.

10.5.2.5 SYSCLK Status Register (SYSTAT)

The SYSCLK Status Register (SYSTAT) shows the status of SYSCLK[4:1]. SYSTAT is shown in Figure 10-12 and described in Table 10-20.

Figure 10-12. SYSCLK Status Register (SYSTAT)

31	4	3	2	1	0
Reserved		SYS4ON	SYS3ON	SYS2ON	SYS1ON
R-n		R-1	R-1	R-1	R-1

Legend: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-20. SYSCLK Status Register Field Descriptions

Bit	Field	Description
31-4	Reserved	Reserved. This location is always read as 0. A value written to this field has no effect.
3-0	SYS[N(1)]ON	SYSCLK[N] on status • 0 = SYSCLK[N] is gated • 1 = SYSCLK[N] is on

(1) Where N = 1, 2, 3, or 4

10.5.2.6 Reset Type Status Register (RSTYPE)

The Reset Type Status (RSTYPE) Register latches the cause of the last reset. If multiple reset sources occur simultaneously, this register latches the highest priority reset source. The Reset Type Status Register is shown in Figure 10-13 and described in Table 10-21.

Figure 10-13. Reset Type Status Register (RSTYPE)

31	29	28	27	12	11	8	7	3	2	1	0
Reserved		EMU-RST	Reserved		WDRST[N]		Reserved		PLLCTRLRST	$\overline{\text{RESET}}$	POR
R-0		R-0	R-0		R-0		R-0		R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-21. Reset Type Status Register Field Descriptions

Bit	Field	Description
31-29	Reserved	Reserved. Always reads as 0. Writes have no effect.
28	EMU-RST	Reset initiated by emulation • 0 = Not the last reset to occur • 1 = The last reset to occur
27-12	Reserved	Reserved. Always reads as 0. Writes have no effect.

Table 10-21. Reset Type Status Register Field Descriptions (continued)

Bit	Field	Description
11 10 9 8	WDRST3 WDRST2 WDRST1 WDRST0	Reset initiated by Watchdog Timer[N] • 0 = Not the last reset to occur • 1 = The last reset to occur
7-3	Reserved	Reserved. Always reads as 0. Writes have no effect.
2	PLLCTLRST	Reset initiated by PLLCTL • 0 = Not the last reset to occur • 1 = The last reset to occur
1	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$ reset • 0 = $\overline{\text{RESET}}$ was not the last reset to occur • 1 = $\overline{\text{RESET}}$ was the last reset to occur
0	POR	Power-on reset • 0 = Power-on reset was not the last reset to occur • 1 = Power-on reset was the last reset to occur

10.5.2.7 Reset Control Register (RSTCTRL)

This register contains a key that enables writes to the MSB of this register and the RSTCFG register. The key value is 0x5A69. A valid key will be stored as 0x000C. Any other key value is invalid. When the RSTCTRL or the RSTCFG is written, the key is invalidated. Every write must be set up with a valid key. The Software Reset Control Register (RSTCTRL) is shown in Figure 10-14 and described in Table 10-22.

Figure 10-14. Reset Control Register (RSTCTRL)

31	17	16	15	0
Reserved		SWRST	KEY	
R-0x0000		R/W-0x(1)	R/W-0x0003	

Legend: R = Read only; -*n* = value after reset;

(1) Writes are conditional based on valid key.

Table 10-22. Reset Control Register Field Descriptions

Bit	Field	Description
31-17	Reserved	Reserved
16	SWRST	Software reset • 0 = Reset • 1 = Not reset
15-0	KEY	Key used to enable writes to RSTCTRL and RSTCFG.

10.5.2.8 Reset Configuration Register (RSTCFG)

This register is used to configure the type of reset (a hard reset or a soft reset) initiated by $\overline{\text{RESET}}$, the watchdog timer, and the Core PLL Controller's RSTCTRL Register. By default, these resets are hard resets. The Reset Configuration Register (RSTCFG) is shown in Figure 10-15 and described in Table 10-23.

Figure 10-15. Reset Configuration Register (RSTCFG)

31	14	13	12	11	4	3	0
Reserved		PLLCTLRSTTYPE	$\overline{\text{RESET}}$TYPE	Reserved		WDTYPE[N(1)]	
R-0x000000		R/W-0(2)	R/W-0(2)	R-0x0		R/W-0x00(2)	

Legend: R = Read only; R/W = Read/Write; -*n* = value after reset

(1) Where N = 1, 2, 3,....N (Not all these outputs may be used on a specific device.)
(2) Writes are conditional based on valid key. For details, see Section 10.5.2.7.

Table 10-23. Reset Configuration Register Field Descriptions

Bit	Field	Description
31-14	Reserved	Reserved
13	PLLCTLRSTTYPE	PLL controller initiates a software-driven reset of type: • 0 = Hard reset (default) • 1 = Soft reset
12	$\overline{\text{RESET}}$TYPE	$\overline{\text{RESET}}$ initiates a reset of type: • 0 = Hard reset (default) • 1 = Soft reset
11-4	Reserved	Reserved
3 2 1 0	WDTYPE3 WDTYPE2 WDTYPE1 WDTYPE0	Watchdog timer [N] initiates a reset of type: • 0 = Hard reset (default) • 1 = Soft reset

10.5.2.9 Reset Isolation Register (RSISO)

This register is used to select the module clocks that must maintain their clocking without pausing through non-power-on reset. Setting any of these bits effectively blocks reset to all Core PLL Control Registers in order to maintain current values of PLL multiplier, divide ratios, and other settings. Along with setting the module-specific bit in RSISO, the corresponding MDCTLx[12] bit also needs to be set in the PSC to reset-isolate a particular module. For more information on the MDCTLx Register, see the *KeyStone Architecture Power Sleep Controller (PSC) User's Guide* (SPRUGV4). The Reset Isolation Register (RSISO) is shown in Figure 10-16 and described in Table 10-24.

Figure 10-16. Reset Isolation Register (RSISO)

31	9	8	7	0
Reserved		SRISO	Reserved	
R-0		R/W-0	R-0	

Legend: R = Read only; R/W = Read/Write; -*n* = value after reset

Table 10-24. Reset Isolation Register Field Descriptions

Bit	Field	Description
31-9	Reserved	Reserved.
8	SRISO	Isolate SmartReflex control • 0 = Not reset isolated • 1 = Reset isolated
7-0	Reserved	Reserved

10.5.3 Core PLL Control Registers

The Core PLL uses two chip-level registers (COREPLLCTL0 and COREPLLCTL1) along with the Core PLL Controller for its configuration. These MMRs (memory-mapped registers) exist inside the Bootcfg space. To write to these registers, software should go through an unlocking sequence using the KICK0 and KICK1 registers. These registers reset only on a $\overline{\text{POR}}$ reset.

For valid configurable values of the COREPLLCTL registers, see Section 8.1.4. See Section 8.2.3.4 for the address location of the KICK registers and their locking and unlocking sequences.

See Figure 10-17 and Table 10-25 for COREPLLCTL0 details and Figure 10-18 and Table 10-26 for COREPLLCTL1 details.

Figure 10-17. Core PLL Control Register 0 (COREPLLCTL0)

31 24	23 19	18 12	11 6	5 0
BWADJ[7:0]	Reserved	PLLM[12:6]	Reserved	PLLD
RW,+0000 0101	RW - 0000 0	RW,+0000000	RW, +000000	RW,+000000

Legend: RW = Read/Write; -*n* = value after reset

Table 10-25. Core PLL Control Register 0 (COREPLLCTL0) Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in COREPLLCTL0 and COREPLLCTL1 registers. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1)>>1) - 1.
23-19	Reserved	Reserved
18-12	PLLM[12:6]	7 bits of a 13-bit field PLLM that selects the values for the multiplication factor. PLLM field is loaded with the multiply factor minus 1. The PLLM[5:0] bits of the multiplier are controlled by the PLLM register inside the PLL Controller and the PLLM[12:6] bits are controlled by the above chip-level register. COREPLLCTL0 register PLLM[12:6] bits should be written just before writing to PLLM register PLLM[5:0] bits in the controller to have the complete 13 bit value latched when the GO operation is initiated in the PLL controller. See the *KeyStone Architecture Phase Locked Loop (PLL) Controller User's Guide* (SPRUGV2) for the recommended programming sequence. Output Divide ratio and Bypass enable/disable of the Core PLL is also controlled by the SECCTL register in the PLL Controller. See the Section 10.5.2.1for more details.
11-6	Reserved	Reserved
5-0	PLLD	A 6-bit field that selects the values for the reference divider. PLLD field is loaded with reference divide value minus 1.

Figure 10-18. Core PLL Control Register 1 (COREPLLCTL1)

31 7	6	5 4	3 0
Reserved	ENSAT	Reserved	BWADJ[11:8]
RW - 00000000000000000000000000	RW-0	R-00	RW- 0000

Legend: RW = Read/Write; -*n* = value after reset

Table 10-26. Core PLL Control Register 1 (COREPLLCTL1) Field Descriptions

Bit	Field	Description
31-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper PLL operation
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in COREPLLCTL0 and COREPLLCTL1 registers. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1)>>1) - 1.

10.5.4 Core PLL Controller/SGMII/XFI/TSREF/HyperLink/PCIe/USB Clock Input Electrical Data/Timing

Table 10-27. Core PLL Controller/SGMII/XFI/TSREF/HyperLink/PCIe/USB Clock Input Timing Requirements(1)

(see Figure 10-19 through Figure 10-21)

NO.			MIN	MAX	UNIT
CORECLK[P:N]					
1	tc(CORECLKN)	Cycle time CORECLKN cycle time	3.2	25	ns
1	tc(CORECLKP)	Cycle time CORECLKP cycle time	3.2	25	ns
3	tw(CORECLKN)	Pulse width CORECLKN high	0.45*tc	0.55*tc	ns
2	tw(CORECLKN)	Pulse width CORECLKN low	0.45*tc	0.55*tc	ns
2	tw(CORECLKP)	Pulse width CORECLKP high	0.45*tc	0.55*tc	ns
3	tw(CORECLKP)	Pulse width CORECLKP low	0.45*tc	0.55*tc	ns
4	tr(CORECLK_200 mV)	Transition time CORECLK differential rise time (200 mV)	50	350	ps
4	tf(CORECLK_200 mV)	Transition time CORECLK differential fall time (200 mV)	50	350	ps
5	tj(CORECLKN)	Jitter, peak_to_peak _ periodic CORECLKN		0.02*tc(CORECLKN)	ps
5	tj(CORECLKP)	Jitter, peak_to_peak _ periodic CORECLKP		0.02*tc(CORECLKP)	ps
SGMII0CLK[P:N]					
1	tc(SGMII0CLKN)	Cycle time SGMII0CLKN cycle time	3.2 or 6.4 or 8		ns
1	tc(SGMII0CLKP)	Cycle time SGMII0CLKP cycle time	3.2 or 6.4 or 8		ns
3	tw(SGMII0CLKN)	Pulse width SGMII0CLKN high	0.45*tc(SGMII0CLKN)	0.55*tc(SGMII0CLKN)	ns
2	tw(SGMII0CLKN)	Pulse width SGMII0CLKN low	0.45*tc(SGMII0CLKN)	0.55*tc(SGMII0CLKN)	ns
2	tw(SGMII0CLKP)	Pulse width SGMII0CLKP high	0.45*tc(SGMII0CLKP)	0.55*tc(SGMII0CLKP)	ns
3	tw(SGMII0CLKP)	Pulse width SGMII0CLKP low	0.45*tc(SGMII0CLKP)	0.55*tc(SGMII0CLKP)	ns
4	tr(SGMII0CLK_200mV)	Transition time SGMII0CLK differential rise time (200 mV)	50	350	ps
4	tf(SGMII0CLK_200mV)	Transition time SGMII0CLK differential fall time (200 mV)	50	350	ps
5	tj(SGMII0CLKN)	Jitter, RMS SGMII0CLKN		4	ps, RMS
5	tj(SGMII0CLKP)	Jitter, RMS SGMII0CLKP		4	ps, RMS
XFICLK[P:N]					
1	tc(XFICLKN)	Cycle time XFICLKN cycle time	3.2 or 6.4		ns
1	tc(XFICLKP)	Cycle time XFICLKP cycle time	3.2 or 6.4		ns
3	tw(XFICLKN)	Pulse width XFICLKN high	0.45*tc(XFICLKN)	0.55*tc(XFICLKN)	ns
2	tw(XFICLKN)	Pulse width XFICLKN low	0.45*tc(XFICLKN)	0.55*tc(XFICLKN)	ns
2	tw(XFICLKP)	Pulse width XFICLKP high	0.45*tc(XFICLKP)	0.55*tc(XFICLKP)	ns
3	tw(XFICLKP)	Pulse width XFICLKP low	0.45*tc(XFICLKP)	0.55*tc(XFICLKP)	ns
4	tr(XFICLK_200mV)	Transition time XFICLK differential rise time (200 mV)	50	350	ps
4	tf(XFICLK_200mV)	Transition time XFICLK differential fall time (200 mV)	50	350	ps
5	tj(XFICLKN)	Jitter, RMS XFICLKN		4	ps, RMS
5	tj(XFICLKP)	Jitter, RMS XFICLKP		4	ps, RMS
HYPLNK0CLK[P:N]					

(1) See the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0) for detailed recommendations.

Table 10-27. Core PLL Controller/SGMII/XFI/TSREF/HyperLink/PCIe/USB Clock Input Timing Requirements[1] (continued)

(see Figure 10-19 through Figure 10-21)

NO.			MIN	MAX	UNIT
1	tc(HYPLNK0CLKN)	Cycle time HYPLNK0CLKN cycle time	3.2 or 6.4		ns
1	tc(HYPLNK0CLKP)	Cycle time HYPLNK0CLKP cycle time	3.2 or 6.4		ns
3	tw(HYPLNK0CLKN)	Pulse width HYPLNK0CLKN high	0.45*tc(HYPLNK0CLKN)	0.55*tc(HYPLNK0CLKN)	ns
2	tw(HYPLNK0CLKN)	Pulse width HYPLNK0CLKN low	0.45*tc(HYPLNK0CLKN)	0.55*tc(HYPLNK0CLKN)	ns
2	tw(HYPLNK0CLKP)	Pulse width HYPLNK0CLKP high	0.45*tc(HYPLNK0CLKP)	0.55*tc(HYPLNK0CLKP)	ns
3	tw(HYPLNK0CLKP)	Pulse width HYPLNK0CLKP low	0.45*tc(HYPLNK0CLKP)	0.55*tc(HYPLNK0CLKP)	ns
4	tr(HYPLNK0CLK)	Rise time HYPLNK0CLK differential rise time (10% to 90%)		0.2*tc(HYPLNK0CLKP)	ps
4	tf(HYPLNK0CLK)	Fall time HYPLNK0CLK differential fall time (10% to 90%)		0.2*tc(HYPLNK0CLKP)	ps
5	tj(HYPLNK0CLKN)	Jitter, RMS HYPLNK0CLKN		4	ps, RMS
5	tj(HYPLNK0CLKP)	Jitter, RMS HYPLNK0CLKP		4	ps, RMS
PCIECLK[P:N]					
1	tc(PCIECLKN)	Cycle time PCIECLKN cycle time	10	10	ns
1	tc(PCIECLKP)	Cycle time PCIECLKP cycle time	10	10	ns
3	tw(PCIECLKN)	Pulse width PCIECLKN high	0.45*tc(PCIECLKN)	0.55*tc(PCIECLKN)	ns
2	tw(PCIECLKN)	Pulse width PCIECLKN low	0.45*tc(PCIECLKN)	0.55*tc(PCIECLKN)	ns
2	tw(PCIECLKP)	Pulse width PCIECLKP high	0.45*tc(PCIECLKP)	0.55*tc(PCIECLKP)	ns
3	tw(PCIECLKP)	Pulse width PCIECLKP low	0.45*tc(PCIECLKP)	0.55*tc(PCIECLKP)	ns
4	tr(PCIECLK)	Rise time PCIECLK differential rise time (10% to 90%)		0.2*tc(PCIECLKP)	ps
4	tf(PCIECLK)	Fall time PCIECLK differential fall time (10% to 90%)		0.2*tc(PCIECLKP)	ps
5	tj(PCIECLKN)	Jitter, RMS PCIECLKN		4	ps, RMS
5	tj(PCIECLKP)	Jitter, RMS PCIECLKP		4	ps, RMS
USBCLK[P:M]					
1	tc(USBCLKN)	Cycle time USBCLKM cycle time	10	10	ns
1	tc(USBCLKP)	Cycle time USBCLKP cycle time	10	10	ns
3	tw(USBCLKN)	Pulse width USBCLKM high	0.45*tc(USBCLKN)	0.55*tc(USBCLKN)	ns
2	tw(USBCLKN)	Pulse width USBCLKM low	0.45*tc(USBCLKN)	0.55*tc(USBCLKN)	ns
2	tw(USBCLKP)	Pulse width USBCLKP high	0.45*tc(USBCLKP)	0.55*tc(USBCLKP)	ns
3	tw(USBCLKP)	Pulse width USBCLKP low	0.45*tc(USBCLKP)	0.55*tc(USBCLKP)	ns
4	tr(USBCLK)	Rise time USBCLK differential rise time (10% to 90%)	50	350	ps
4	tf(USBCLK)	Fall time USBCLK differential fall time (10% to 90%)	50	350	ps
5	tj(USBCLKN)	Jitter, RMS USBCLKM		4	ps, RMS
5	tj(USBCLKP)	Jitter, RMS USBCLKP		4	ps, RMS
TSREFCLK[P:N][2]					
1	tc(TSREFCLKN)	Cycle time TSREFCLKN cycle time	3.25	32.55	ns
1	tc(TSREFCLKP)	Cycle time TSREFCLKP cycle time	3.25	32.55	ns
3	tw(TSREFCLKN)	Pulse width TSREFCLKN high	0.45*tc(TSREFCLKN)	0.55*tc(TSREFCLKN)	ns

(2) TSREFCLK clock input is LVDS compliant.

Table 10-27. Core PLL Controller/SGMII/XFI/TSREF/HyperLink/PCIe/USB Clock Input Timing Requirements[1] (continued)

(see Figure 10-19 through Figure 10-21)

NO.			MIN	MAX	UNIT
2	tw(TSREFCLKN)	Pulse width TSREFCLKN low	0.45*tc(TSREFCLKN)	0.55*tc(TSREFCLKN)	ns
2	tw(TSREFCLKP)	Pulse width TSREFCLKP high	0.45*tc(TSREFCLKP)	0.55*tc(TSREFCLKP)	ns
3	tw(TSREFCLKP)	Pulse width TSREFCLKP low	0.45*tc(TSREFCLKP)	0.55*tc(TSREFCLKP)	ns
4	tr(TSREFCLK_200mV)	Transition time TSREFCLK differential rise time (200 mV)	50	350	ps
4	tf(TSREFCLK_200mV)	Transition time TSREFCLK differential fall time (200 mV)	50	350	ps
5	tj(TSREFCLKN)	Jitter, RMS TSREFCLKN		5.8	ps, RMS
5	tj(TSREFCLKP)	Jitter, RMS TSREFCLKP		5.8	ps, RMS

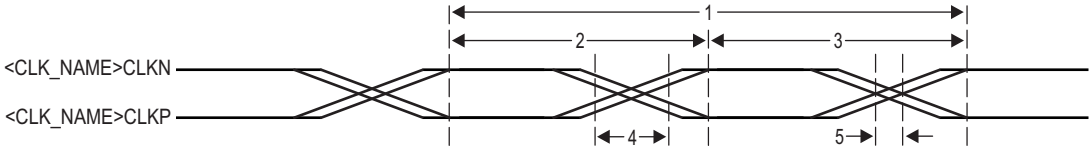


Figure 10-19. Clock Input Timing

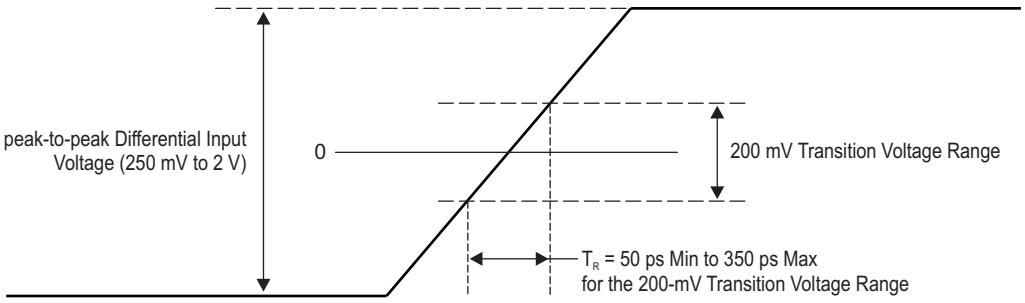


Figure 10-20. CORECLK, SGMII0CLK and USBCLK Clock Transition Time

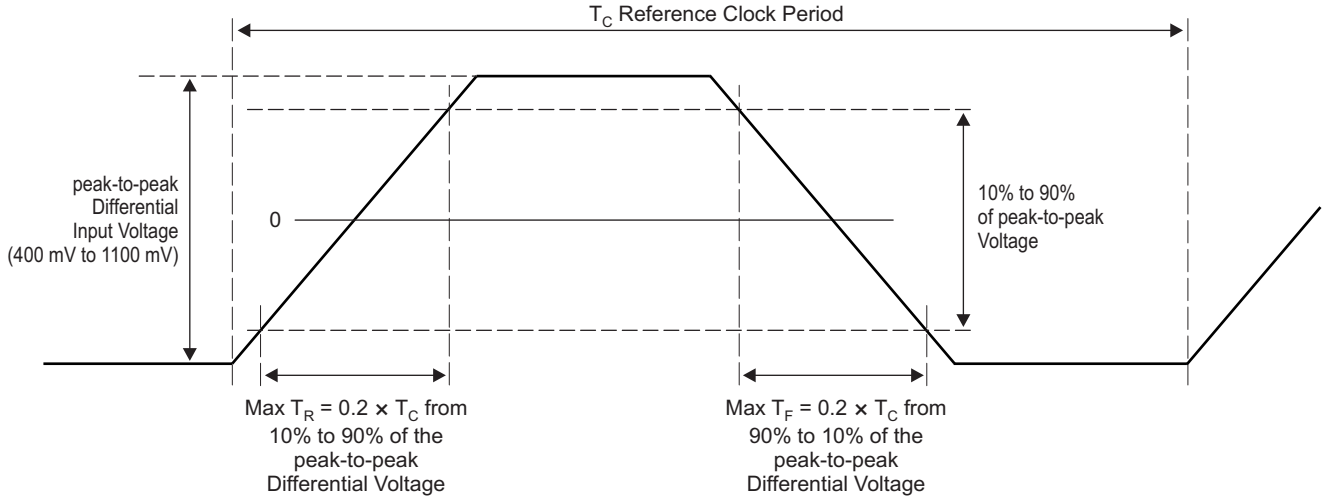


Figure 10-21. HYPLNK0CLK, XFICLK, and PCIECLK Rise and Fall Times

10.6 DDR3 PLL

The DDR3 PLL generates interface clocks for the DDR3 memory controller. When coming out of power-on reset, DDR3 PLL is programmed to a valid frequency during the boot configuration process before being enabled and used.

DDR3 PLL power is supplied via the DDR3 PLL power-supply pin (AVDDA2). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0) for detailed recommendations.

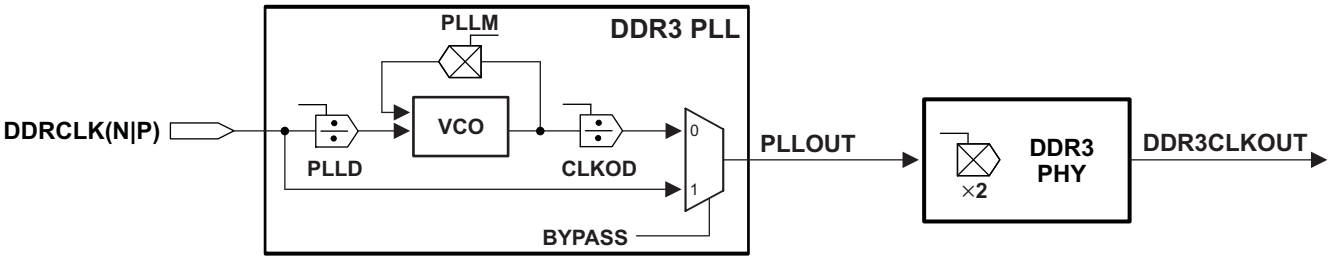


Figure 10-22. DDR3 PLL Block Diagram

10.6.1 DDR3 PLL Control Registers

The DDR3 PLL, which is used to drive the DDR3 PHY for the EMIF, does not use a PLL controller. DDR3 PLL can be controlled using the DDR3PLLCTL0 and DDR3PLLCTL1 registers located in the Bootcfg module. These MMRs (memory-mapped registers) exist inside the Bootcfg space. To write to these registers, software must go through an unlocking sequence using the KICK0 and KICK1 registers. For suggested configurable values, see Section 8.1.4. See Section 8.2.3.4 for the address location of the registers and locking and unlocking sequences for accessing the registers. These registers are reset on POR only.

Figure 10-23. DDR3 PLL Control Register 0 (DDR3PLLCTL0)

31 24	23	22 19	18 6	5 0
BWADJ[7:0]	BYPASS	CLKOD	PLLM	PLLD
RW,+0000 1001	RW,+0	RW,+0001	RW,+0000000010011	RW,+000000

Legend: RW = Read/Write; -*n* = value after reset

Table 10-28. DDR3 PLL Control Register 0 Field Descriptions

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in DDR3PLLCTL0 and DDR3PLLCTL1 registers. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1)>>1) - 1.
23	BYPASS	Enable bypass mode • 0 = Bypass disabled • 1 = Bypass enabled
22-19	CLKOD	A 4-bit field that selects the values for the PLL post divider. Valid post divider values are 1 and even values from 2 to 16. CLKOD field is loaded with output divide value minus 1
18-6	PLLM	A 13-bit field that selects the values for the PLL multiplication factor. PLLM field is loaded with the multiply factor minus 1
5-0	PLLD	A 6-bit field that selects the values for the reference (input) divider. PLLD field is loaded with reference divide value minus 1

Figure 10-24. DDR3 PLL Control Register 1 (DDR3PLLCTL1)

31 15	14	13 7	6	5 4	3 0
Reserved	PLLRST	Reserved	ENSAT	Reserved	BWADJ[11:8]
RW - 00000000000000000	RW-0	RW-0000000	RW-0	R-00	RW- 0000

Legend: RW = Read/Write; -*n* = value after reset

Table 10-29. DDR3 PLL Control Register 1 Field Descriptions

Bit	Field	Description
31-15	Reserved	Reserved
14	PLLRST	PLL Reset bit • 0 = PLL Reset is released • 1 = PLL Reset is asserted
13-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper PLL operation
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in the DDR3PLLCTL0 and the DDR3PLLCTL1 registers. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1)>>1) - 1.

10.6.2 DDR3 PLL Device-Specific Information

As shown in Figure 10-22, the output of DDR3 PLL (PLLOUT) is divided by 2 and directly fed to the DDR3 memory controller. During power-on resets, the internal clocks of the DDR3 PLL are affected as described in Section Section 10.4. The DDR3 PLL is unlocked only during the power-up sequence and is locked by the time the $\overline{\text{RESETSTAT}}$ pin goes high. It does not lose lock during any of the other resets.

10.6.3 DDR3 PLL Input Clock Electrical Data/Timing

Table 10-30 applies to DDR3 memory interface.

Table 10-30. DDR3 PLL DDRCLK(N|P) Timing Requirements

(see Figure 10-25 and Figure 10-20)

No.			Min	Max	Unit
DDRCLK[P:N]					
1	tc(DDRCLKN)	Cycle time _ DDRCLKN cycle time	3.2	25	ns
1	tc(DDRCLKP)	Cycle time _ DDRCLKP cycle time	3.2	25	ns
3	tw(DDRCLKN)	Pulse width _ DDRCLKN high	0.45*tc(DDRCLKN)	0.55*tc(DDRCLKN)	ns
2	tw(DDRCLKN)	Pulse width _ DDRCLKN low	0.45*tc(DDRCLKN)	0.55*tc(DDRCLKN)	ns
2	tw(DDRCLKP)	Pulse width _ DDRCLKP high	0.45*tc(DDRCLKP)	0.55*tc(DDRCLKP)	ns
3	tw(DDRCLKP)	Pulse width _ DDRCLKP low	0.45*tc(DDRCLKP)	0.55*tc(DDRCLKP)	ns
4	tr(DDRCLK_200 mV)	Transition time _ DDRCLK differential rise time (200 mV)	50	350	ps
4	tf(DDRCLK_200 mV)	Transition time _ DDRCLK differential fall time (200 mV)	50	350	ps
5	tj(DDRCLKN)	Jitter, peak_to_peak _ periodic DDRCLKN		0.02*tc(DDRCLKN)	ps
5	tj(DDRCLKP)	Jitter, peak_to_peak _ periodic DDRCLKP		0.02*tc(DDRCLKP)	ps

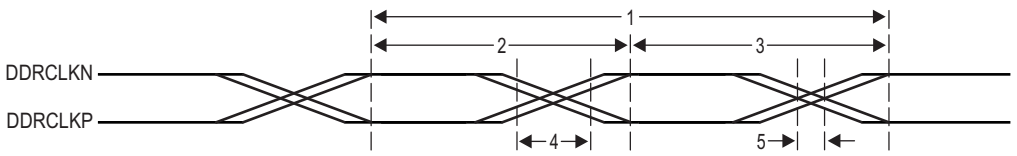


Figure 10-25. DDR3 PLL DDRCLK Timing

10.7 NETCP PLL

The NETCP PLL generates interface clocks for the Network Coprocessor. Using the NETCPCLKSEL pin the user can select the input source of the NETCP PLL as either the output of the Core PLL mux or the NETCPCLK clock reference source. When coming out of power-on reset, NETCP PLL comes out in a bypass mode and needs to be programmed to a valid frequency before being enabled and used.

NETCP PLL power is supplied via the NETCP PLL power-supply pin (AVDDA3). An external EMI filter circuit must be added to all PLL supplies. See the *Hardware Design Guide for KeyStone II Devices* application report (SPRABV0) for detailed recommendations.

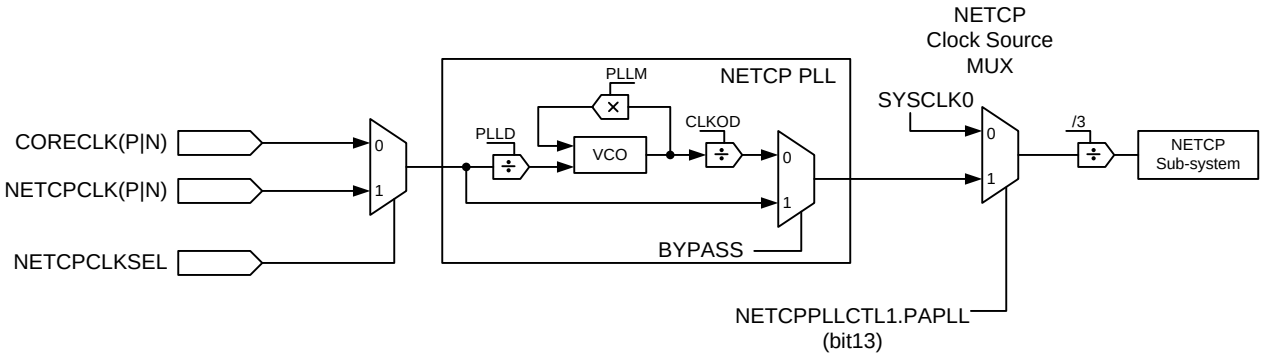


Figure 10-26. NETCP PLL Block Diagram

10.7.1 NETCP PLL Local Clock Dividers

The clock signal from the NETCP PLL Controller is routed to the Network Coprocessor. The NETCP module has two internal dividers with fixed division ratios. See table Table 10-31.

10.7.2 NETCP PLL Control Registers

The NETCP PLL, which is used to drive the Network Coprocessor, does not use a PLL controller. NETCP PLL can be controlled using the NETCPPLLCTL0 and NETCPPLLCTL1 registers located in the Bootcfg module. These MMRs (memory-mapped registers) exist inside the Bootcfg space. To write to these registers, software must go through an unlocking sequence using the KICK0 and KICK1 registers. For suggested configuration values, see Section 8.1.4. See Section 8.2.3.4 for the address location of the registers and locking and unlocking sequences for accessing these registers. These registers are reset on POR only.

Figure 10-27. NETCP PLL Control Register 0 (NETCPPLLCTL0)

31 – 24	23	22 – 19	18 – 6	5 – 0
BWADJ[7:0]	BYPASS	CLKOD	PLLM	PLLD
RW,+0000 1001	RW,+0	RW,+0001	RW,+0000000010011	RW,+000000

Legend: RW = Read/Write; -*n* = value after reset

Table 10-31. NETCP PLL Control Register 0 Field Descriptions (NETCPPLLCTL0)

Bit	Field	Description
31-24	BWADJ[7:0]	BWADJ[11:8] and BWADJ[7:0] are located in NETCPPLLCTL0 and NETCPPLLCTL1 registers. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1)>>1) - 1.
23	BYPASS	Enable bypass mode • 0 = Bypass disabled • 1 = Bypass enabled
22-19	CLKOD	A 4-bit field that selects the values for the PLL post divider. Valid post divider values are 1 and even values from 2 to 16. CLKOD field is loaded with output divide value minus 1

Table 10-31. NETCP PLL Control Register 0 Field Descriptions (NETCPPLLCTL0) (continued)

Bit	Field	Description
18-6	PLLM	A 13-bit field that selects the values for the multiplication factor. PLLM field is loaded with the multiply factor minus 1.
5-0	PLLD	A 6-bit field that selects the values for the reference divider. PLLD field is loaded with reference divide value minus 1.

Figure 10-28. NETCP PLL Control Register 1 (NETCPPLLCTL1)

31 15	14	13	12 7	6	5 4	3 0
Reserved	PLLRST	PAPLL	Reserved	ENSAT	Reserved	BWADJ[11:8]
RW - 00000000000000000	RW-0	RW-0	RW-000000	RW-0	R-00	RW-0000

Legend: RW = Read/Write; -*n* = value after reset

Table 10-32. NETCP PLL Control Register 1 Field Descriptions (NETCPPLLCTL1)

Bit	Field	Description
31-15	Reserved	Reserved
14	PLLRST	PLL Reset bit • 0 = PLL Reset is released • 1 = PLL Reset is asserted
13	PAPLL	NETCP Clock Source MUX Control • 0 = SYSCLK0 • 1 = NETCP PLL
12-7	Reserved	Reserved
6	ENSAT	Needs to be set to 1 for proper PLL operation
5-4	Reserved	Reserved
3-0	BWADJ[11:8]	BWADJ[11:8] and BWADJ[7:0] are located in NETCPPLLCTL0 and NETCPPLLCTL1 registers. BWADJ[11:0] should be programmed to a value related to PLLM[12:0] value based on the equation: BWADJ = ((PLLM+1)>>1) - 1.

10.7.3 NETCP PLL Device-Specific Information

As shown in Figure 10-26, the output of NETCP PLL (PLLOUT) is divided by 3 and directly fed to the Network Coprocessor. During power-on resets, the internal clocks of the NETCP PLL are affected as described in Section 10.4. The NETCP PLL is unlocked only during the power-up sequence and is locked by the time the RESETSTAT pin goes high. It does not lose lock during any other resets.

10.7.4 NETCP PLL Input Clock Electrical Data/Timing

Table 10-33. NETCP PLL Timing Requirements

(see Figure 10-29 and Figure 10-20)

NO.			MIN	MAX	UNIT
		NETCPCLK[P:N]			
1	tc(NETCPCLKN)	Cycle time _ NETCPCLKN cycle time	3.2	25	ns
1	tc(NETCPCLKP)	Cycle time _ NETCPCLKP cycle time	3.2	25	ns
3	tw(NETCPCLKN)	Pulse width _ NETCPCLKN high	0.45*tc(NETCPCLKN)	0.55*tc(NETCPCLKN)	ns
2	tw(NETCPCLKN)	Pulse width _ NETCPCLKN low	0.45*tc(NETCPCLKN)	0.55*tc(NETCPCLKN)	ns
2	tw(NETCPCLKP)	Pulse width _ NETCPCLKP high	0.45*tc(NETCPCLKP)	0.55*tc(NETCPCLKP)	ns
3	tw(NETCPCLKP)	Pulse width _ NETCPCLKP low	0.45*tc(NETCPCLKP)	0.55*tc(NETCPCLKP)	ns
4	tr(NETCPCLK_250mV)	Transition time _ NETCPCLK differential rise time (250 mV)	50	350	ps
4	tf(NETCPCLK_250mV)	Transition time _ NETCPCLK differential fall time (250 mV)	50	350	ps
5	tj(NETCPCLKN)	Jitter, peak_to_peak _ periodic NETCPCLKN		100	ps, pk-pk
5	tj(NETCPCLKP)	Jitter, peak_to_peak _ periodic NETCPCLKP		100	ps, pk-pk

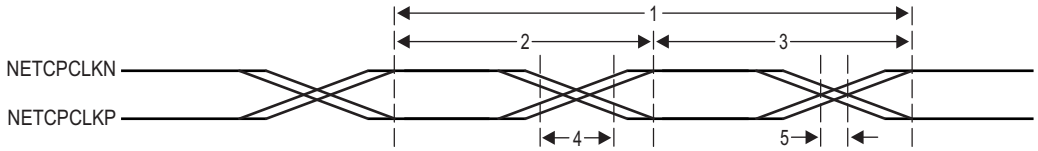


Figure 10-29. NETCP PLL Timing

10.8 DDR3 Memory Controller

The 72-bit DDR3 Memory Controller bus of the AM5K2E0x is used to interface to JEDEC standard-compliant DDR3 SDRAM devices. The DDR3 external bus interfaces only to DDR3 SDRAM devices and does not share the bus with any other type of peripheral.

10.8.1 DDR3 Memory Controller Device-Specific Information

The AM5K2E0x includes one 64-bit wide, 1.5-V DDR3 SDRAM EMIF interface. The DDR3 interface can operate at 800 mega transfers per second (MTS), 1033 MTS, 1333 MTS, and 1600 MTS.

Due to the complicated nature of the interface, a limited number of topologies are supported to provide a 16-bit, 32-bit, or 64-bit interface.

The DDR3 electrical requirements are fully specified in the DDR JEDEC Specification JESD79-3C. Standard DDR3 SDRAMs are available in 8-bit and 16-bit versions allowing for the following bank topologies to be supported by the interface:

- **72-bit:** Five 16-bit SDRAMs (including 8 bits of ECC)
- **72-bit:** Nine 8-bit SDRAMs (including 8 bits of ECC)
- **36-bit:** Three 16-bit SDRAMs (including 4 bits of ECC)
- **36-bit:**Five 8-bit SDRAMs (including 4 bits of ECC)
- **64-bit:**Four 16-bit SDRAMs
- **64-bit:**Eight 8-bit SDRAMs
- **32-bit:**Two 16-bit SDRAMs
- **32-bit:** Four 8-bit SDRAMs
- **16-bit:**One 16-bit SDRAM
- **16-bit:**Two 8-bit SDRAMs

The approach to specifying interface timing for the DDR3 memory bus is different than on other interfaces such as I^2C or SPI. For these other interfaces, the device timing was specified in terms of data manual specifications and I/O buffer information specification (IBIS) models. For the DDR3 memory bus, the approach is to specify compatible DDR3 devices and provide the printed circuit board (PCB) solution and guidelines directly to the user.

A race condition may exist when certain masters write data to the DDR3 memory controller. For example, if master A passes a software message via a buffer in external memory and does not wait for an indication that the write completes before signaling to master B that the message is ready, when master B attempts to read the software message, the master B read may bypass the master A write. Thus, master B may read stale data and receive an incorrect message.

Some master peripherals (e.g., EDMA3 transfer controllers with TCCMOD=0) always wait for the write to complete before signaling an interrupt to the system, thus avoiding this race condition. For masters that do not have a hardware specification of write-read ordering, it may be necessary to specify data ordering in the software.

If master A does not wait for an indication that a write is complete, it must perform the following workaround:

1. Perform the required write to DDR3 memory space.
2. Perform a dummy write to the DDR3 memory controller module ID and revision register.

3. Perform a dummy read to the DDR3 memory controller module ID and revision register.
4. Indicate to master B that the data is ready to be read after completion of the read in step 3. The completion of the read in step 3 ensures that the previous write was done.

10.8.2 DDR3 Slew Rate Control

The DDR3 slew rate is controlled by use of the PHY registers. See the*KeyStone Architecture DDR3 Memory Controller User's Guide* SPRUGV8 for details.

10.8.3 DDR3 Memory Controller Electrical Data/Timing

The *DDR3 Design Requirements for KeyStone Devices* application report SPRABI1 specifies a complete DDR3 interface solution as well as a list of compatible DDR3 devices. The DDR3 electrical requirements are fully specified in the DDR3 JEDEC Specification JESD79-3C. TI has performed the simulation and system characterization to ensure all DDR3 interface timings in this solution are met. Therefore, no electrical data/timing information is supplied here for this interface.

NOTE

TI supports ***only*** designs that follow the board design guidelines outlined in the application report.

10.9 I²C Peripheral

The Inter-Integrated Circuit (I^2C) module provides an interface between SoC and other devices compliant with Philips Semiconductors (now NXP Semiconductors) Inter-Integrated Circuit bus specification version 2.1. External components attached to this 2-wire serial bus can transmit/receive up to 8-bit data to/from the device through the I^2C module.

10.9.1 I²C Device-Specific Information

The device includes multiple I^2C peripheral modules.

NOTE

When using the I^2C module, ensure there are external pullup resistors on the SDA and SCL pins.

The I^2C modules on the AM5K2E0x may be used by the SoC to control local peripheral ICs (DACs, ADCs, etc.), communicate with other controllers in a system, or to implement a user interface.

The I^2C port supports:

- Compatibility with Philips I^2C specification revision 2.1 (January 2000)
- Fast mode up to 400 kbps (no fail-safe I/O buffers)
- Noise filter to remove noise of 50 ns or less
- 7-bit and 10-bit device addressing modes
- Multi-master (transmit/receive) and slave (transmit/receive) functionality
- Events: DMA, interrupt, or polling
- Slew-rate limited open-drain output buffers

Figure 10-30 shows a block diagram of the I^2C module.

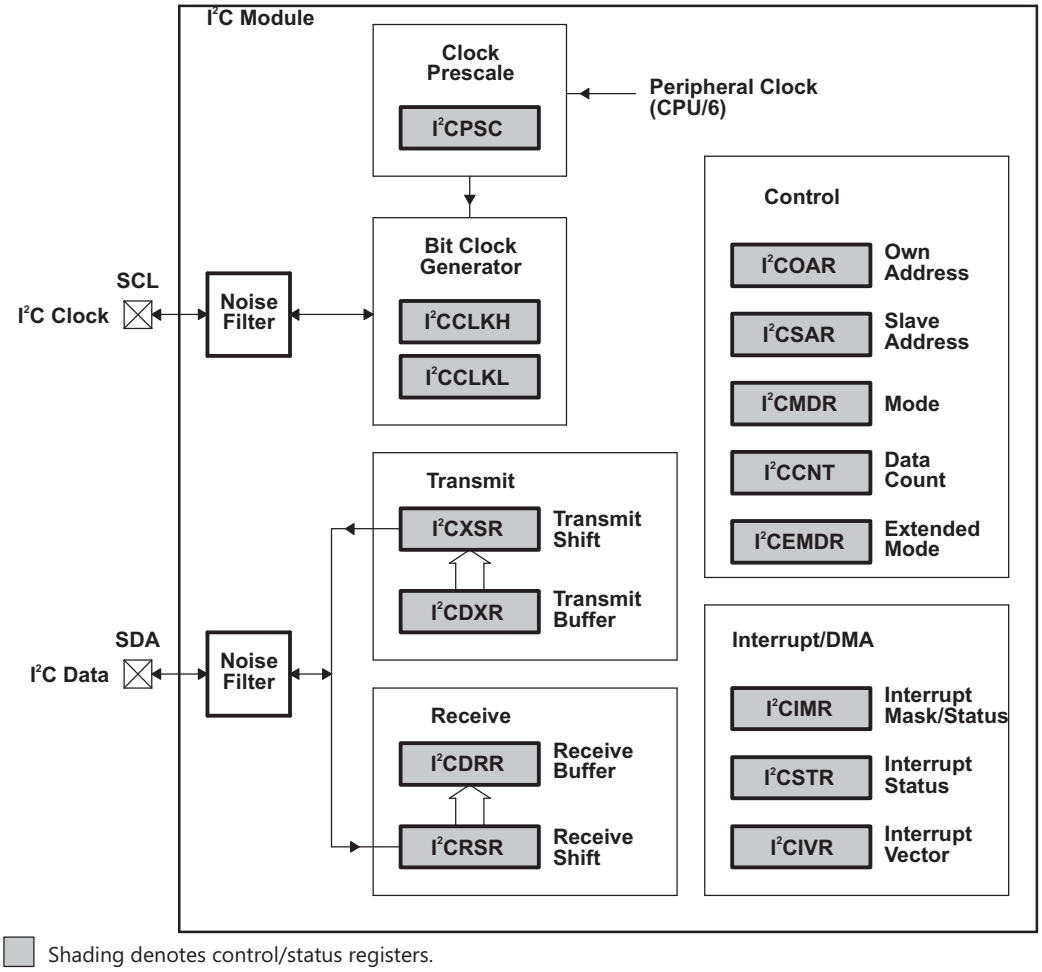


Figure 10-30. I²C Module Block Diagram

10.9.2 I²C Peripheral Register Description

Table 10-34. I²C Registers

HEX ADDRESS OFFSETS	ACRONYM	REGISTER NAME
0x0000	ICOAR	I²C Own Address Register
0x0004	ICIMR	I²C Interrupt Mask/status Register
0x0008	ICSTR	I²C Interrupt Status Register
0x000C	ICCLKL	I²C Clock Low-time Divider Register
0x0010	ICCLKH	I²C Clock High-time Divider Register
0x0014	ICCNT	I²C Data Count Register
0x0018	ICDRR	I²C Data Receive Register
0x001C	ICSAR	I²C Slave Address Register
0x0020	ICDXR	I²C Data Transmit Register
0x0024	ICMDR	I²C Mode Register
0x0028	ICIVR	I²C Interrupt Vector Register
0x002C	ICEMDR	I²C Extended Mode Register
0x0030	ICPSC	I²C Prescaler Register
0x0034	ICPID1	I²C Peripheral Identification Register 1 [value: 0x0000 0105]
0x0038	ICPID2	I²C Peripheral Identification Register 2 [value: 0x0000 0005]

Table 10-34. I²C Registers (continued)

HEX ADDRESS OFFSETS	ACRONYM	REGISTER NAME
0x003C -0x007F	-	Reserved

10.9.3 I²C Electrical Data/Timing

10.9.3.1 Inter-Integrated Circuits (I²C) Timing

Table 10-35. I²C Timing Requirements (1)

(see Figure 10-31)

NO.			STANDARD MODE		FAST MODE		UNIT
			MIN	MAX	MIN	MAX	
1	$t_{c(SCL)}$	Cycle time, SCL	10		2.5		µs
2	$t_{su(SCLH-SDAL)}$	Setup time, SCL high before SDA low (for a repeated START condition)	4.7		0.6		µs
3	$t_{h(SDAL-SCLL)}$	Hold time, SCL low after SDA low (for a START and a repeated START condition)	4		0.6		µs
4	$t_{w(SCLL)}$	Pulse duration, SCL low	4.7		1.3		µs
5	$t_{w(SCLH)}$	Pulse duration, SCL high	4		0.6		µs
6	$t_{su(SDAV-SCLH)}$	Setup time, SDA valid before SCL high	250		100 (2)		ns
7	$t_{h(SCLL-SDAV)}$	Hold time, SDA valid after SCL low (for I²C bus devices)	0 (3)	3.45	0 (3)	0.9 (4)	µs
8	$t_{w(SDAH)}$	Pulse duration, SDA high between STOP and START conditions	4.7		1.3		µs
9	$t_{r(SDA)}$	Rise time, SDA		1000	20 + 0.1C_b (5)	300	ns
10	$t_{r(SCL)}$	Rise time, SCL		1000	20 + 0.1C_b (5)	300	ns
11	$t_{f(SDA)}$	Fall time, SDA		300	20 + 0.1C_b (5)	300	ns
12	$t_{f(SCL)}$	Fall time, SCL		300	20 + 0.1C_b (5)	300	ns
13	$t_{su(SCLH-SDAH)}$	Setup time, SCL high before SDA high (for STOP condition)	4		0.6		µs
14	$t_{w(SP)}$	Pulse duration, spike (must be suppressed)			0	50	ns
	C_b (5)	Capacitive load for each bus line		400		400	pF

(1) The I²C pins SDA and SCL do not feature fail-safe I/O buffers. These pins could potentially draw current when the device is powered down.
(2) A Fast-mode I²C-bus device can be used in a Standard-mode I²C-bus system, but the requirement tsu(SDA-SCLH) ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line t_r max + $t_{su(SDA-SCLH)}$ = 1000 + 250 = 1250 ns (according to the Standard-mode I²C-Bus Specification) before the SCL line is released.
(3) A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHmin} of the SCL signal) to bridge the undefined region of the falling edge of SCL.
(4) The maximum $t_{h(SDA-SCLL)}$ has to be met only if the device does not stretch the low period [$t_{w(SCLL)}$] of the SCL signal.
(5) C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

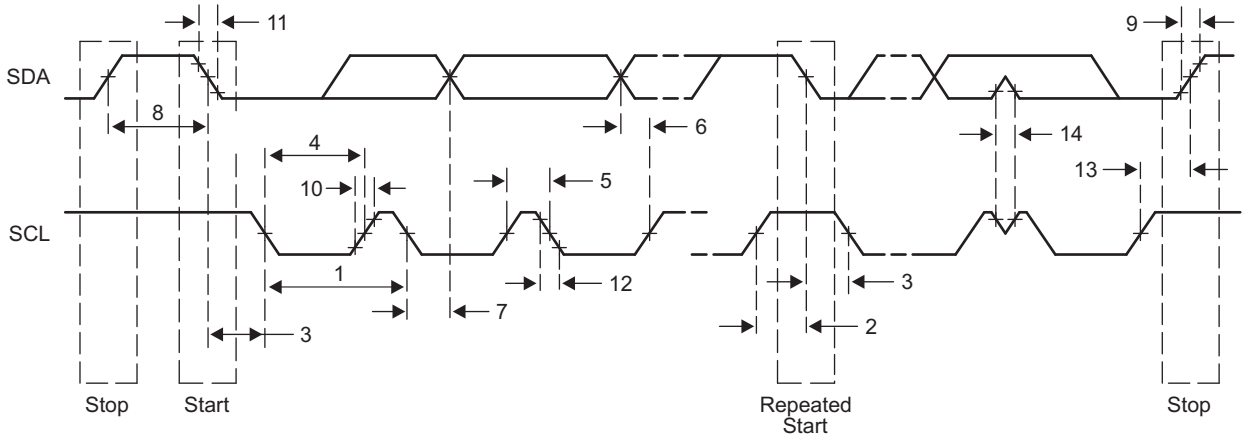


Figure 10-31. I²C Receive Timings

Table 10-36. I²C Switching Characteristics(1)

(see Figure 10-32)

NO.	PARAMETER		STANDARD MODE		FAST MODE		UNIT
			MIN	MAX	MIN	MAX	
16	$t_{c(SCL)}$	Cycle time, SCL	10		2.5		µs
17	$t_{su(SCLH-SDAL)}$	Setup time, SCL high to SDA low (for a repeated START condition)	4.7		0.6		µs
18	$t_{h(SDAL-SCLL)}$	Hold time, SDA low after SCL low (for a START and a repeated START condition)	4		0.6		µs
19	$t_{w(SCLL)}$	Pulse duration, SCL low	4.7		1.3		µs
20	$t_{w(SCLH)}$	Pulse duration, SCL high	4		0.6		µs
21	$t_{d(SDAV-SDLH)}$	Delay time, SDA valid to SCL high	250		100		ns
22	$t_{v(SDLL-SDAV)}$	Valid time, SDA valid after SCL low (for I²C bus devices)	0		0	0.9	µs
23	$t_{w(SDAH)}$	Pulse duration, SDA high between STOP and START conditions	4.7		1.3		µs
24	$t_{r(SDA)}$	Rise time, SDA		1000	$20 + 0.1C_b$ (1)	300	ns
25	$t_{r(SCL)}$	Rise time, SCL		1000	$20 + 0.1C_b$ (1)	300	ns
26	$t_{f(SDA)}$	Fall time, SDA		300	$20 + 0.1C_b$ (1)	300	ns
27	$t_{f(SCL)}$	Fall time, SCL		300	$20 + 0.1C_b$ (1)	300	ns
28	$t_{d(SCLH-SDAH)}$	Delay time, SCL high to SDA high (for STOP condition)	4		0.6		µs
	C_p	Capacitance for each I²C pin		10		10	pF

(1) C_b = total capacitance of one bus line in pF. If mixed with HS-mode devices, faster fall-times are allowed.

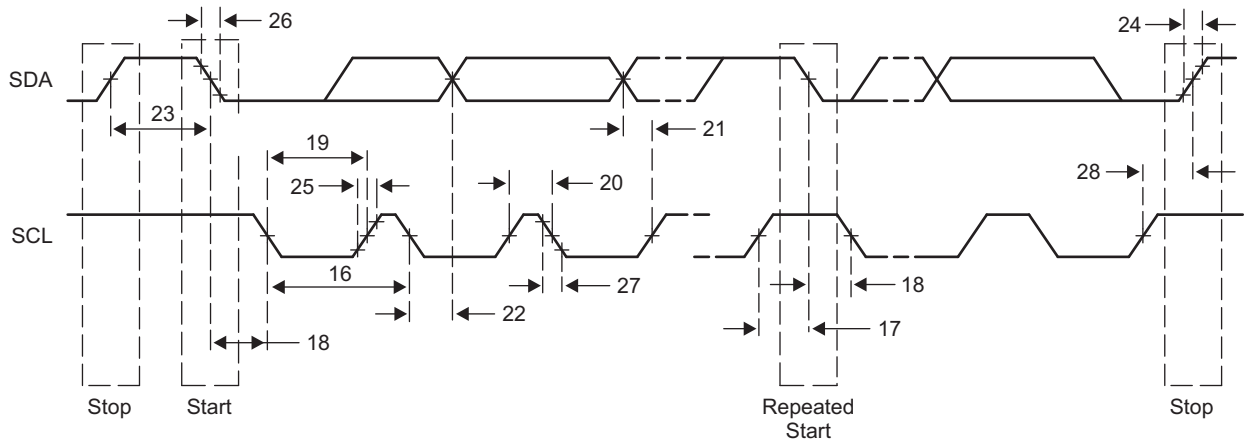


Figure 10-32. I²C Transmit Timings

10.10 SPI Peripheral

The Serial Peripheral Interconnect (SPI) module provides an interface between the SoC and other SPI-compliant devices. The primary intent of this interface is to allow for connection to an SPI ROM for boot. The SPI module on AM5K2E0x is supported only in master mode. Additional chip-level components can also be included, such as temperature sensors or an I/O expander.

10.10.1 SPI Electrical Data/Timing

Table 10-37. SPI Timing Requirements

(see Figure 10-33)

NO.			MIN	MAX	UNIT
Master Mode Timing Diagrams — Base Timings for 3 Pin Mode					
7	tsu(SPIDIN-SPC)	Input setup time, SPIDIN valid before receive edge of SPICLK. Polarity = 0 Phase = 0	2		ns
7	tsu(SPIDIN-SPC)	Input setup time, SPIDIN valid before receive edge of SPICLK. Polarity = 0 Phase = 1	2		ns
7	tsu(SPIDIN-SPC)	Input setup time, SPIDIN valid before receive edge of SPICLK. Polarity = 1 Phase = 0	2		ns
7	tsu(SPIDIN-SPC)	Input setup time, SPIDIN valid before receive edge of SPICLK. Polarity = 1 Phase = 1	2		ns
8	th(SPC-SPIDIN)	Input hold time, SPIDIN valid after receive edge of SPICLK. Polarity = 0 Phase = 0	5		ns
8	th(SPC-SPIDIN)	Input hold time, SPIDIN valid after receive edge of SPICLK. Polarity = 0 Phase = 1	5		ns
8	th(SPC-SPIDIN)	Input hold time, SPIDIN valid after receive edge of SPICLK. Polarity = 1 Phase = 0	5		ns
8	th(SPC-SPIDIN)	Input hold time, SPIDIN valid after receive edge of SPICLK. Polarity = 1 Phase = 1	5		ns

Table 10-38. SPI Switching Characteristics

(see Figure 10-33 and Figure 10-34)

NO.	PARAMETER		MIN	MAX	UNIT
Master Mode Timing Diagrams — Base Timings for 3 Pin Mode					
1	tc(SPC)	Cycle time, SPICLK, all master modes	3*P2(1)		ns
2	tw(SPCH)	Pulse width high, SPICLK, all master modes	0.5*(3*P2) - 1		ns
3	tw(SPCL)	Pulse width low, SPICLK, all master modes	0.5*(3*P2) - 1		ns
4	td(SPIDOUT-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 0, Phase = 0.		5	ns
4	td(SPIDOUT-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK. Polarity = 0, Phase = 1.		5	ns
4	td(SPIDOUT-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK Polarity = 1, Phase = 0		5	ns

(1) P2=1/(SYSCLK1/6)

Table 10-38. SPI Switching Characteristics (continued)

(see Figure 10-33 and Figure 10-34)

NO.	PARAMETER		MIN	MAX	UNIT
4	td(SPIDOUT-SPC)	Setup (Delay), initial data bit valid on SPIDOUT to initial edge on SPICLK Polarity = 1, Phase = 1		5	ns
5	td(SPC-SPIDOUT)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK. Polarity = 0 Phase = 0		2	ns
5	td(SPC-SPIDOUT)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK Polarity = 0 Phase = 1		2	ns
5	td(SPC-SPIDOUT)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK Polarity = 1 Phase = 0		2	ns
5	td(SPC-SPIDOUT)	Setup (Delay), subsequent data bits valid on SPIDOUT to initial edge on SPICLK Polarity = 1 Phase = 1		2	ns
6	toh(SPC-SPIDOUT)	Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 0 Phase = 0	0.5*tc - 2		ns
6	toh(SPC-SPIDOUT)	Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 0 Phase = 1	0.5*tc - 2		ns
6	toh(SPC-SPIDOUT)	Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 1 Phase = 0	0.5*tc - 2		ns
6	toh(SPC-SPIDOUT)	Output hold time, SPIDOUT valid after receive edge of SPICLK except for final bit. Polarity = 1 Phase = 1	0.5*tc - 2		ns
Additional SPI Master Timings — 4 Pin Mode with Chip Select Option					
19	td(SCS-SPC)	Delay from SPISCSx\ active to first SPICLK. Polarity = 0 Phase = 0	2*P2 - 5	2*P2 + 5	ns
19	td(SCS-SPC)	Delay from SPISCSx\ active to first SPICLK. Polarity = 0 Phase = 1	0.5*tc + (2*P2) - 5	0.5*tc + (2*P2) + 5	ns
19	td(SCS-SPC)	Delay from SPISCSx\ active to first SPICLK. Polarity = 1 Phase = 0	2*P2 - 5	2*P2 + 5	ns
19	td(SCS-SPC)	Delay from SPISCSx\ active to first SPICLK. Polarity = 1 Phase = 1	0.5*tc + (2*P2) - 5	0.5*tc + (2*P2) + 5	ns
20	td(SPC-SCS)	Delay from final SPICLK edge to master deasserting SPISCSx\. Polarity = 0 Phase = 0	1*P2 - 5	1*P2 + 5	ns
20	td(SPC-SCS)	Delay from final SPICLK edge to master deasserting SPISCSx\. Polarity = 0 Phase = 1	0.5*tc + (1*P2) - 5	0.5*tc + (1*P2) + 5	ns
20	td(SPC-SCS)	Delay from final SPICLK edge to master deasserting SPISCSx\. Polarity = 1 Phase = 0	1*P2 - 5	1*P2 + 5	ns
20	td(SPC-SCS)	Delay from final SPICLK edge to master deasserting SPISCSx\. Polarity = 1 Phase = 1	0.5*tc + (1*P2) - 5	0.5*tc + (1*P2) + 5	ns
	tw(SCSH)	Minimum inactive time on SPISCSx\ pin between two transfers when SPISCSx\ is not held using the CSHOLD feature.	2*P2 - 5		ns

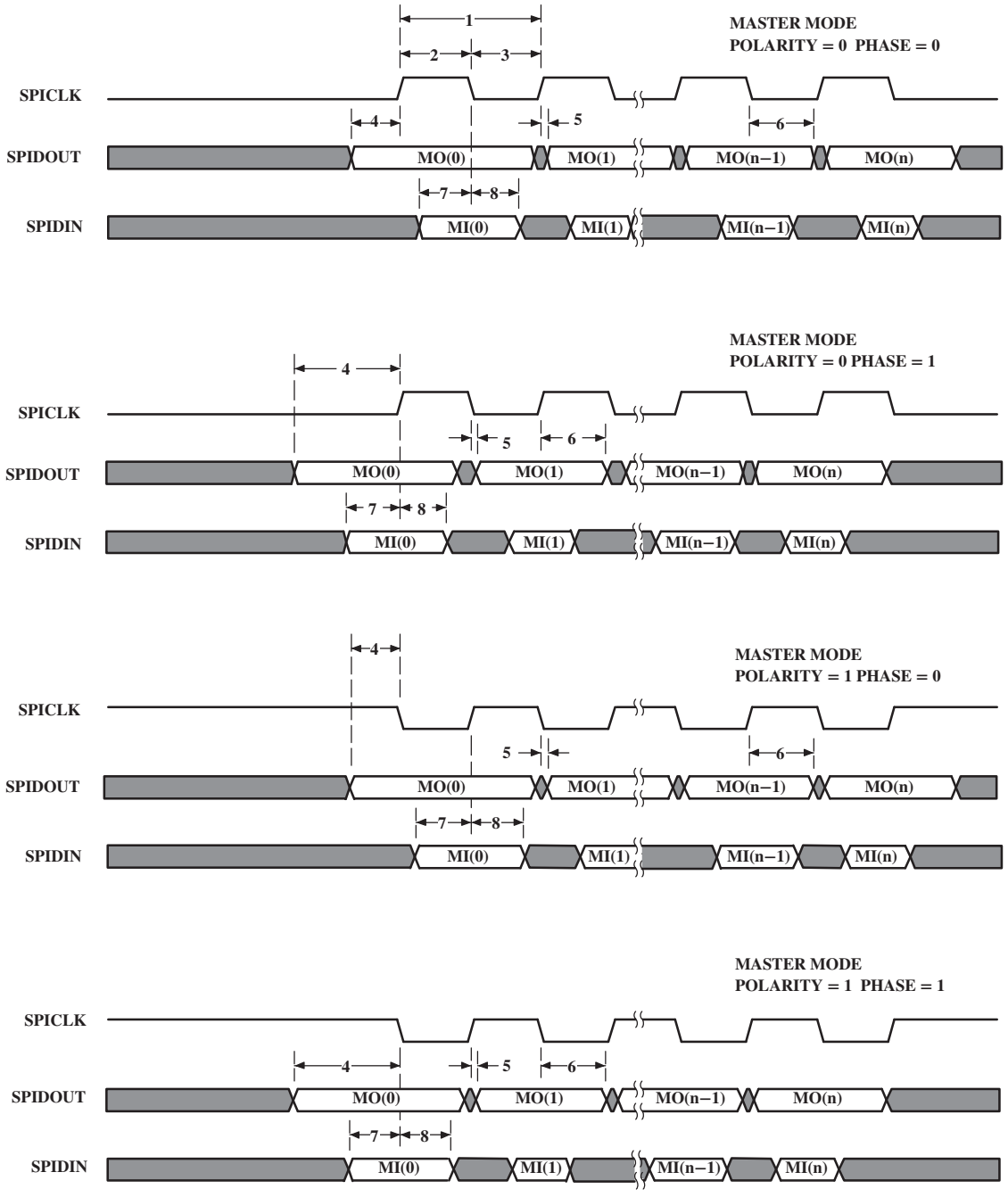


Figure 10-33. SPI Master Mode Timing Diagrams — Base Timings for 3-Pin Mode

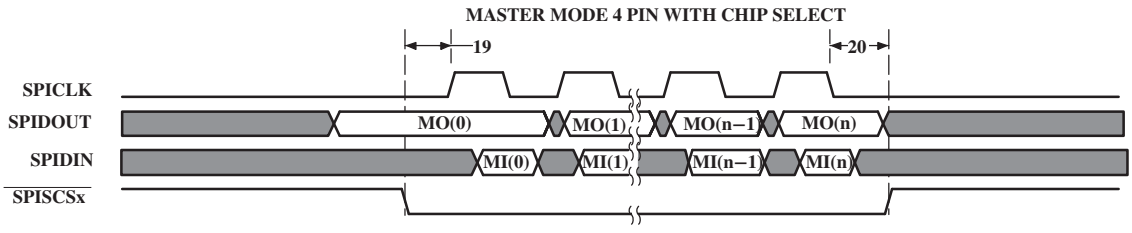


Figure 10-34. SPI Additional Timings for 4-Pin Master Mode with Chip Select Option

10.11 HyperLink Peripheral

The AM5K2E0x includes HyperLink for companion device interfaces. This is a four-lane SerDes interface designed to operate at up to 10 Gbps per lane from pin-to-pin. The interface is used to connect with external accelerators that are manufactured using TI libraries. The HyperLink lines must be connected with DC coupling.

The interface includes the serial station management interfaces used to send power management and flow messages between devices. Each HyperLink interface consists of four LVCMOS inputs and four LVCMOS outputs configured as two 2-wire input buses and two 2-wire output buses. Each 2-wire bus includes a data signal and a clock signal.

Table 10-39. HyperLink Peripheral Timing Requirements

(see Figure 10-35, Figure 10-36 and Figure 10-37)

NO.			MIN	MAX	UNIT
FL Interface					
1	tc(HYPTXFLCLK)	Clock period - HYPTXFLCLK (C1)	5.75		ns
2	tw(HYPTXFLCLKH)	High pulse width - HYPTXFLCLK	0.4*C1	0.6*C1	ns
3	tw(HYPTXFLCLKL)	Low pulse width - HYPTXFLCLK	0.4*C1	0.6*C1	ns
6	tsu(HYPTXFLDAT-HYPTXFLCLKH)	Setup time - HYPTXFLDAT valid before HYPTXFLCLK high	1		ns
7	th(HYPTXFLCLKH-HYPTXFLDAT)	Hold time - HYPTXFLDAT valid after HYPTXFLCLK high	1		ns
6	tsu(HYPTXFLDAT-HYPTXFLCLKL)	Setup time - HYPTXFLDAT valid before HYPTXFLCLK low	1		ns
7	th(HYPTXFLCLKL-HYPTXFLDAT)	Hold time - HYPTXFLDAT valid after HYPTXFLCLK low	1		ns
PM Interface					
1	tc(HYPRXPMCLK)	Clock period - HYPRXPMCLK (C3)	5.75		ns
2	tw(HYPRXPMCLK)	High pulse width - HYPRXPMCLK	0.4*C3	0.6*C3	ns
3	tw(HYPRXPMCLK)	Low pulse width - HYPRXPMCLK	0.4*C3	0.6*C3	ns
6	tsu(HYPRXPMDAT-HYPRXPMCLKH)	Setup time - HYPRXPMDAT valid before HYPRXPMCLK high	1		ns
7	th(HYPRXPMCLKH-HYPRXPMDAT)	Hold time - HYPRXPMDAT valid after HYPRXPMCLK high	1		ns
6	tsu(HYPRXPMDAT-HYPRXPMCLKL)	Setup time - HYPRXPMDAT valid before HYPRXPMCLK low	1		ns
7	th(HYPRXPMCLKL-HYPRXPMDAT)	Hold time - HYPRXPMDAT valid after HYPRXPMCLK low	1		ns

Table 10-40. HyperLink Peripheral Switching Characteristics

(see Figure 10-35, Figure 10-36 and Figure 10-37)

NO.	PARAMETER		MIN	MAX	UNIT
FL Interface					
1	tc(HYPRXFLCLK)	Clock period - HYPRXFLCLK (C2)	6.4		ns
2	tw(HYPRXFLCLKH)	High pulse width - HYPRXFLCLK	0.4*C2	0.6*C2	ns
3	tw(HYPRXFLCLKL)	Low pulse width - HYPRXFLCLK	0.4*C2	0.6*C2	ns
4	tosu(HYPRXFLDAT-HYPRXFLCLKH)	Setup time - HYPRXFLDAT valid before HYPRXFLCLK high	0.25*C2-0.4		ns
5	toh(HYPRXFLCLKH-HYPRXFLDAT)	Hold time - HYPRXFLDAT valid after HYPRXFLCLK high	0.25*C2-0.4		ns
4	tosu(HYPRXFLDAT-HYPRXFLCLKL)	Setup time - HYPRXFLDAT valid before HYPRXFLCLK low	0.25*C2-0.4		ns
5	toh(HYPRXFLCLKL-HYPRXFLDAT)	Hold time - HYPRXFLDAT valid after HYPRXFLCLK low	0.25*C2-0.4		ns
PM Interface					
1	tc(HYPTXPMCLK)	Clock period - HYPTXPMCLK (C4)	6.4		ns
2	tw(HYPTXPMCLK)	High pulse width - HYPTXPMCLK	0.4*C4	0.6*C4	ns
3	tw(HYPTXPMCLK)	Low pulse width - HYPTXPMCLK	0.4*C4	0.6*C4	ns
4	tosu(HYPTXPMDAT-HYPTXPMCLKH)	Setup time - HYPTXPMDAT valid before HYPTXPMCLK high	0.25*C2-0.4		ns

Table 10-40. HyperLink Peripheral Switching Characteristics (continued)

(see Figure 10-35, Figure 10-36 and Figure 10-37)

NO.	PARAMETER		MIN	MAX	UNIT
5	toh(HYPTXPMCLKH-HYPTXPMDAT)	Hold time - HYPTXPMDAT valid after HYPTXPMCLK high	0.25*C2-0.4		ns
4	tosu(HYPTXPMDAT-HYPTXPMCLKL)	Setup time - HYPTXPMDAT valid before HYPTXPMCLK low	0.25*C2-0.4		ns
5	toh(HYPTXPMCLKL-HYPTXPMDAT)	Hold time - HYPTXPMDAT valid after HYPTXPMCLK low	0.25*C2-0.4		ns

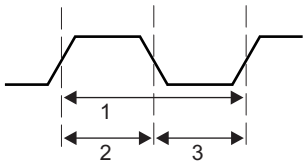
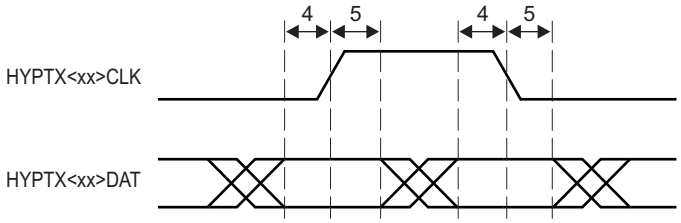
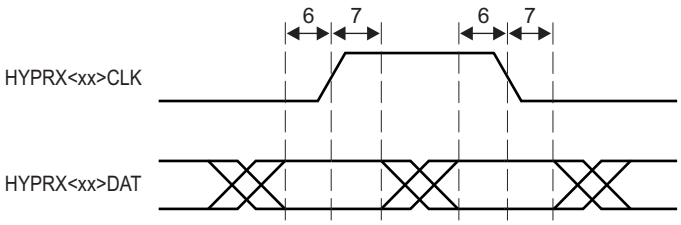


Figure 10-35. HyperLink Station Management Clock Timing



<xx> represents the interface that is being used: PM or FL

Figure 10-36. HyperLink Station Management Transmit Timing



<xx> represents the interface that is being used: PM or FL

Figure 10-37. HyperLink Station Management Receive Timing

10.12 UART Peripheral

The universal asynchronous receiver/transmitter (UART) module provides an interface between the device and a UART terminal interface or other UART-based peripheral. The UART is based on the industry standard TL16C550 asynchronous communications element which, in turn, is a functional upgrade of the TL16C450. Functionally similar to the TL16C450 on power up (single character or TL16C450 mode), the UART can be placed in an alternate FIFO (TL16C550) mode. This relieves the SoC of excessive software overhead by buffering received and transmitted characters. The receiver and transmitter FIFOs store up to 16 bytes including three additional bits of error status per byte for the receiver FIFO.

The UART performs serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the SoC to be sent to the peripheral device. The SoC can read the UART status at any time. The UART includes control capability and a processor interrupt system that can be tailored to minimize software management of the communications link. For more information on UART, see the *KeyStone Architecture Universal Asynchronous Receiver/Transmitter (UART) User's Guide* (SPRUGP1).

Table 10-41. UART Timing Requirements

(see Figure 10-38 and Figure 10-39)

NO.			MIN	MAX	UNIT
Receive Timing					
4	tw(RXSTART)	Pulse width, receive start bit	0.96U[1]	1.05U	ns
5	tw(RXH)	Pulse width, receive data/parity bit high	0.96U	1.05U	ns
5	tw(RXL)	Pulse width, receive data/parity bit low	0.96U	1.05U	ns
6	tw(RXSTOP1)	Pulse width, receive stop bit 1	0.96U	1.05U	ns
6	tw(RXSTOP15)	Pulse width, receive stop bit 1.5	0.96U	1.05U	ns
6	tw(RXSTOP2)	Pulse width, receive stop bit 2	0.96U	1.05U	ns
Autoflow Timing Requirements					
8	td(CTSL-TX)	Delay time, CTS asserted to START bit transmit	P[2]	5P	ns

(1) U = UART baud time = 1/programmed baud rate
(2) P = 1/(SYSCLK1/6)

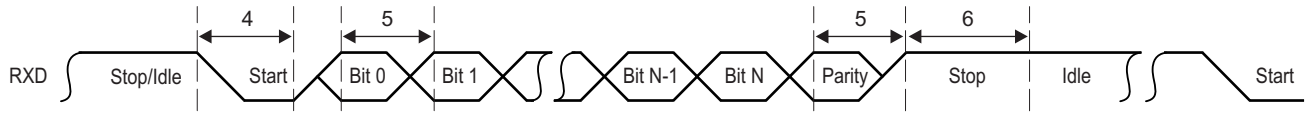


Figure 10-38. UART Receive Timing Waveform

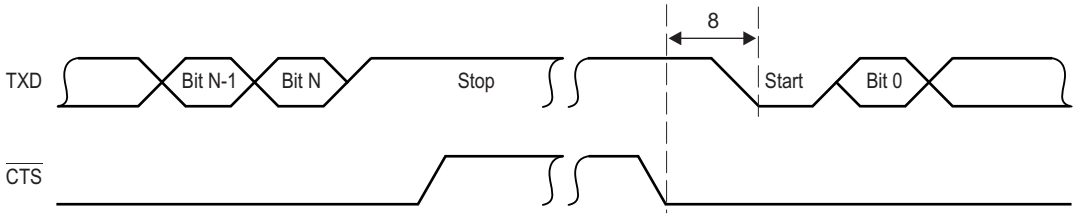


Figure 10-39. UART CTS (Clear-to-Send Input) — Autoflow Timing Waveform

Table 10-42. UART Switching Characteristics

(see Figure 10-40 and Figure 10-41)

NO.	PARAMETER		MIN	MAX	UNIT
Transmit Timing					
1	tw(TXSTART)	Pulse width, transmit start bit	U(1)- 2	U + 2	ns
2	tw(TXH)	Pulse width, transmit data/parity bit high	U - 2	U + 2	ns
2	tw(TXL)	Pulse width, transmit data/parity bit low	U - 2	U + 2	ns
3	tw(TXSTOP1)	Pulse width, transmit stop bit 1	U - 2	U + 2	ns
3	tw(TXSTOP15)	Pulse width, transmit stop bit 1.5	1.5 * (U - 2)	1.5 * ('U + 2)	ns
3	tw(TXSTOP2)	Pulse width, transmit stop bit 2	2 * (U - 2)	2 * ('U + 2)	ns
Autoflow Timing Requirements					
7	td(RX-RTSH)	Delay time, STOP bit received to RTS deasserted	P(2)	5P	ns

(1) U = UART baud time = 1/programmed baud rate
(2) P = 1/(SYSCLK1/6)

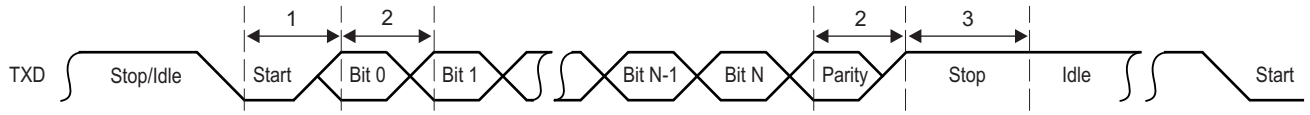


Figure 10-40. UART Transmit Timing Waveform

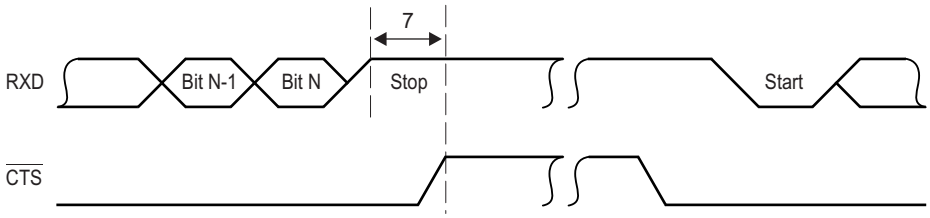


Figure 10-41. UART RTS (Request-to-Send Output) – Autoflow Timing Waveform

10.13 PCIe Peripheral

The two-lane PCI express (PCIe) module on AM5K2E0x provides an interface between the device and other PCIe-compliant devices. The PCIe module provides low pin-count, high-reliability, and high-speed data transfer at rates up to 5.0 Gbps per lane on the serial links. For more information, see the *KeyStone Architecture Peripheral Component Interconnect Express (PCIe) User's Guide* (SPRUGS6).

10.14 Packet Accelerator

The Packet Accelerator (PA) provides L2 to L4 classification functionalities and supports classification for Ethernet, VLAN, MPLS over Ethernet, IPv4/6, GRE over IP, and other session identification over IP such as UDP ports. It maintains 8k multiple-in, multiple-out hardware queues and also provides checksum capability as well as some QoS capabilities. The PA enables a single IP address to be used for a multicore device and can process up to 1.5 Mpps. The Packet Accelerator is coupled with the Network Coprocessor. For more information, see the *KeyStone II Architecture Packet Accelerator 2 (PA2) for K2E and K2L Devices User's Guide* (SPRUHZ2).

10.15 Security Accelerator

The Security Accelerator (SA) provides wire-speed processing on 1 Gbps Ethernet traffic on IPSec, SRTP, and 3GPP Air interface security protocols. It functions on the packet level with the packet and the associated security context being one of the above three types. The Security Accelerator is coupled with the Network Coprocessor, and receives the packet descriptor containing the security context in the buffer descriptor and the data to be encrypted/decrypted in the linked buffer descriptor. For more information, see the *KeyStone II Architecture Security Accelerator 2 (SA2) for K2E and K2L Devices User's Guide* (SPRUHZ1).

10.16 Network Coprocessor Gigabit Ethernet (GbE) Switch Subsystem

The gigabit Ethernet (GbE) switch subsystem provides an efficient interface between the device and the networked community. The Ethernet Media Access Controller (EMAC) supports 10Base-T (10 Mbits/second), and 100BaseTX (100 Mbps), in half- or full-duplex mode, and 1000BaseT (1000 Mbps) in full-duplex mode, with hardware flow control and quality-of-service (QOS) support. The GbE switch subsystem is coupled with the Network Coprocessor. For more information, see the *Gigabit Ethernet (GbE) Switch Subsystem (1 GB) User's Guide* (SPRUGV9).

An address range is assigned to the AM5K2E0x. Each individual device has a 48-bit MAC address and consumes only one unique MAC address out of the range. There are two registers to hold these values, MACID1[31:0] (32 bits) and MACID2[15:0] (16 bits) . The bits of these registers are defined as follows:

Figure 10-42. MACID1 Register (MMR Address 0x02620110)

31	0
MACID	
R,+xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx	

Legend: R = Read only; -x, value is indeterminate

Table 10-43. MACID1 Register Field Descriptions

Bit	Field	Description
31-0	MAC ID	MAC ID. Lower 32 bits.

Figure 10-43. MACID2 Register (MMR Address 0x02620114)

31	24	23	18	17	16	15	0
CRC		Reserved		FLOW	BCAST	MACID	
R+,cccc cccc		R,+rr rrrr		R,+z	R,+y	R,+xxxx xxxx xxxx xxxx	

LEGEND: R = Read only; -x = value is indeterminate

Table 10-44. MACID2 Register Field Descriptions

Bit	Field	Description
31-24	Reserved	Variable
23-18	Reserved	000000
17	FLOW	MAC Flow Control • 0 = Off • 1 = On
16	BCAST	Default m/b-cast reception • 0 = Broadcast • 1 = Disabled
15-0	MAC ID	MAC ID. Upper 16 bits.

There is a central processor time synchronization (CPTS) submodule in the Ethernet switch module that can be used for time synchronization. Programming this register selects the clock source for the CPTS_RCLK. See the *Gigabit Ethernet (GbE) Switch Subsystem (1 GB) User's Guide* (SPRUGV9) for the register address and other details about the time synchronization submodule. The register CPTS_RFTCLK_SEL for reference clock selection of the time synchronization submodule is shown in Figure 10-44.

CPTS also allows 8 HW signal inputs for timestamping. Two of these signals are connected to TSPUSHEVT0 and TSPUSHEVT1. The other 6 are connected to internal SyncE and timer signals. See Table 10-45 for interconnectivity. Regarding the SyncE signal, see Section 8.2.3.23 for more details on how to control this input. Furthermore, see the *Gigabit Ethernet (GbE) Switch Subsystem (1 GB) User's Guide* (SPRUGV9) for details on how to enable HW timestamping on CPTS.

Table 10-45. CPTS Hardware Push Events

EVENT NUMBER	CONNECTION
1	syncE
2	XGE sync
3	Tspushevt1
4	Tspushevt0
5	Timi1
6	Timi0
7	Reserved
8	Reserved

Figure 10-44. RFTCLK Select Register (CPTS_RFTCLK_SEL)

31	4	3	0
Reserved		CPTS_RFTCLK_SEL	
R - 0		RW - 0	

Legend: R = Read only; -x, value is indeterminate

Table 10-46. RFTCLK Select Register Field Descriptions

Bit	Field	Description
31-4	Reserved	Reserved. Read as 0.
3-0	CPTS_RFTCLK_SEL	Reference clock select. This signal is used to control an external multiplexer that selects one of 8 clocks for time sync reference (RFTCLK). This CPTS_RFTCLK_SEL value can be written only when the CPTS_EN bit is cleared to 0 in the TS_CTL register. • 0000 = SYSCLK2 • 0001 = SYSCLK3 • 0010 = TIMI0 • 0011 = TIMI1 • 0100 = TSIPCLKA • 1000 = TSREFCLK • 1100 = TSIPCLKB • Others = Reserved

10.17 SGMII/XFI Management Data Input/Output (MDIO)

The management data input/output (MDIO) module implements the 802.3 serial management interface to interrogate and control up to 32 Ethernet PHY(s) connected to the device, using a shared two-wire bus. Application software uses the MDIO module to configure the auto-negotiation parameters of each PHY attached to the EMAC, retrieve the negotiation results, and configure required parameters in the gigabit Ethernet (GbE) and 10-gigabit Ethernet (10GbE) switch subsystems for correct operation. The module allows almost transparent operation of the MDIO interface, with very little attention from the SoC. For more information, see the *Gigabit Ethernet (GbE) Switch Subsystem (1 GB) User's Guide* (SPRUGV9) and the *KeyStone II Architecture 10 Gigabit Ethernet Subsystem User's Guide* (SPRUHJ5).

Table 10-47. MDIO Timing Requirements

(see Figure 10-45)

NO.			MIN	MAX	UNIT
1	tc(MDCLK)	Cycle time, MDCLK	400		ns
2	tw(MDCLKH)	Pulse duration, MDCLK high	180		ns
3	tw(MDCLKL)	Pulse duration, MDCLK low	180		ns
4	tsu(MDIO-MDCLKH)	Setup time, MDIO data input valid before MDCLK high	10		ns
5	th(MDCLKH-MDIO)	Hold time, MDIO data input valid after MDCLK high	10		ns
	tt(MDCLK)	Transition time, MDCLK		5	ns

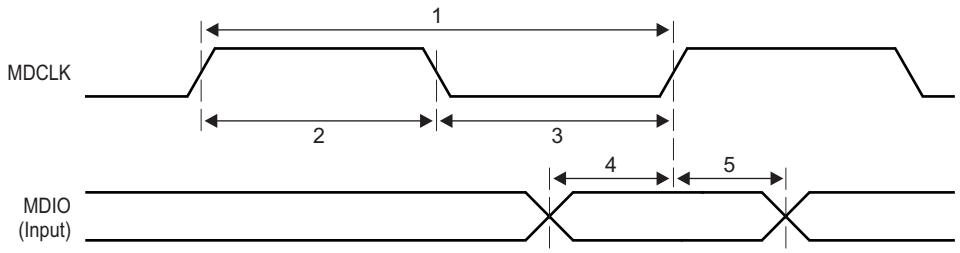


Figure 10-45. MDIO Input Timing

Table 10-48. MDIO Switching Characteristics

(see Figure 10-46)

NO.	PARAMETER		MIN	MAX	UNIT
6	td(MDCLKH-MDIO)	Delay time, MDCLK high to MDIO data output valid	10	300	ns
7	th(MDCLKH-MDIO)	Hold time, MDIO data output valid after MDCLK high	10		ns
8	td(MDCLKH-MDIO)	Delay time, MDCLK high to MDIO Hi-Z	10	300	ns

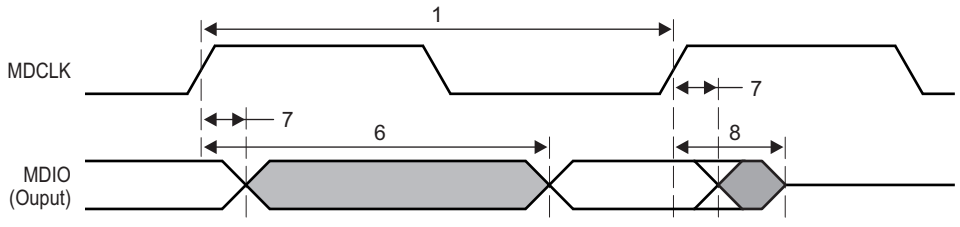


Figure 10-46. MDIO Output Timing

10.18 Ten-Gigabit Ethernet (10GbE) Switch Subsystem

The 3-port Ten Gigabit Ethernet Switch Subsystem (different from the Network Coprocessor integrated switch) includes a standalone EMAC switch subsystem and a 2-lane SerDes macro. The 2-lane macro enables only 2 external ports. It does not include any packet acceleration or security acceleration engine.

10.18.1 10GbE Supported Features

The key features of the 10GbE module are listed below:

- 10 Gbps EMAC switch subsystem
  - MDIO: Media-dependent input/output module
  - SGMII Interface for 10/100/1000 and 10GBASE-KR for 10G
  - Ethernet switch with wire-rate switching (only two external ports are supported by the SerDes)
  - CPTS module that supports time-stamping for IEEE1588v2 with support for eight hardware push events and generation of compare output pulses
  - Supports XFI electrical interface
- CPDMA

The CPDMA component provides CPPI 4.2 compatible functionality, and provides a 128-bit wide data path to the TeraNet, enabling:

- Support for 8 transmit channel and 16 receive channels
- Support for reset isolation option

For more information, see the *KeyStone II Architecture 10 Gigabit Ethernet Subsystem User's Guide* (SPRUHJ5).

10.19 Timers

The timers can be used to time events, count events, generate pulses, interrupt the ARM CorePac and send synchronization events to the EDMA3 channel controller.

10.19.1 Timers Device-Specific Information

The AM5K2E0x device has up to twenty 64-bit timers in total, but only 12 timers are used in AM5K2E04 and 10 timers are used in AM5K2E02, of which Timer16 and Timer17 (AM5K2E02) and Timer16 through Timer19 (AM5K2E04) are dedicated to each of the Cortex-A15 processor cores as a watchdog timer and can also be used as general-purpose timers. The Timer8 through Timer15 can be configured as general-purpose timers only, with each timer programmed as a 64-bit timer or as two separate 32-bit timers.

When operating in 64-bit mode, the timer counts either module clock cycles or input (TINPLx) pulses (rising edge) and generates an output pulse/waveform (TOUTLx) plus an internal event (TINTLx) on a software-programmable period. When operating in 32-bit mode, the timer is split into two independent 32-bit timers. Each timer is made up of two 32-bit counters: a high counter and a low counter. The timer pins, TINPLx and TOUTLx are connected to the low counter. The timer pins, TINPHx and TOUTHx are connected to the high counter.

When operating in watchdog mode, the timer counts down to 0 and generates an event. It is a requirement that software writes to the timer before the count expires, after which the count begins again. If the count ever reaches 0, the timer event output is asserted. Reset initiated by a watchdog timer can be set by programming the Reset Type Status Register (RSTYPE) (see Section 10.5.2.6) and the type of reset initiated can set by programming the Reset Configuration Register (RSTCFG) (see Section 10.5.2.8). For more information, see the *KeyStone Architecture Timer 64P User's Guide* SPRUGV5.

10.19.2 Timers Electrical Timing

The tables and figures below describe the timing requirements and switching characteristics of the timers.

Table 10-49. Timer Input Timing Requirements(1)

(see Figure 10-47)

NO.			MIN	MAX	UNIT
1	$t_{w(TINPH)}$	Pulse duration, high	12C		ns
2	$t_{w(TINPL)}$	Pulse duration, low	12C		ns

(1) C = 1/SYSCLK1 clock frequency in ns

Table 10-50. Timer Output Switching Characteristics(1)

(see Figure 10-47)

NO.	PARAMETER		MIN	MAX	UNIT
3	$t_{w(TOUTH)}$	Pulse duration, high	12C - 3		ns
4	$t_{w(TOUTL)}$	Pulse duration, low	12C - 3		ns

(1) C = 1/SYSCLK1 clock frequency in ns.

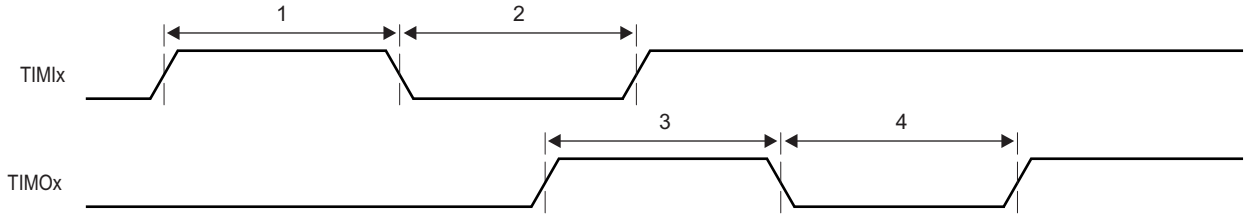


Figure 10-47. Timer Timing

10.20 General-Purpose Input/Output (GPIO)

10.20.1 GPIO Device-Specific Information

The GPIO peripheral pins are used for general purpose input/output for the device. These pins are also used to configure the device at boot time.

For more detailed information on device/peripheral configuration and the AM5K2E0x device pin muxing, see Section 8.2.

These GPIO pins can also be used to generate individual core interrupts (no support of bank interrupt) and EDMA events.

10.20.2 GPIO Peripheral Register Description

Table 10-51. GPIO Registers

Hex Address Offsets	Acronym	Register Name
0x0008	BINTEN	GPIO interrupt per bank enable register
0x000C	-	Reserved
0x0010	DIR	GPIO Direction Register
0x0014	OUT_DATA	GPIO Output Data Register
0x0018	SET_DATA	GPIO Set Data Register
0x001C	CLR_DATA	GPIO Clear Data Register
0x0020	IN_DATA	GPIO Input Data Register
0x0024	SET_RIS_TRIG	GPIO Set Rising Edge Interrupt Register
0x0028	CLR_RIS_TRIG	GPIO Clear Rising Edge Interrupt Register
0x002C	SET_FAL_TRIG	GPIO Set Falling Edge Interrupt Register
0x0030	CLR_FAL_TRIG	GPIO Clear Falling Edge Interrupt Register
0x008C	-	Reserved

Table 10-51. GPIO Registers (continued)

Hex Address Offsets	Acronym	Register Name
0x0090 - 0x03FF	-	Reserved

10.20.3 GPIO Electrical Data/Timing

Table 10-52. GPIO Input Timing Requirements(1)

(see Figure 10-48)

NO.			MIN	MAX	UNIT
1	$t_{w(GPOH)}$	Pulse duration, GPOx high	12C		ns
2	$t_{w(GPOL)}$	Pulse duration, GPOx low	12C		ns

(1) C = 1/SYSCLK1 clock frequency in ns

Table 10-53. GPIO Output Switching Characteristics(1)

(see Figure 10-48)

NO.	PARAMETER		MIN	MAX	UNIT
3	$t_{w(GPOH)}$	Pulse duration, GPOx high	36C - 8		ns
4	$t_{w(GPOL)}$	Pulse duration, GPOx low	36C - 8		ns

(1) C = 1/SYSCLK1 clock frequency in ns

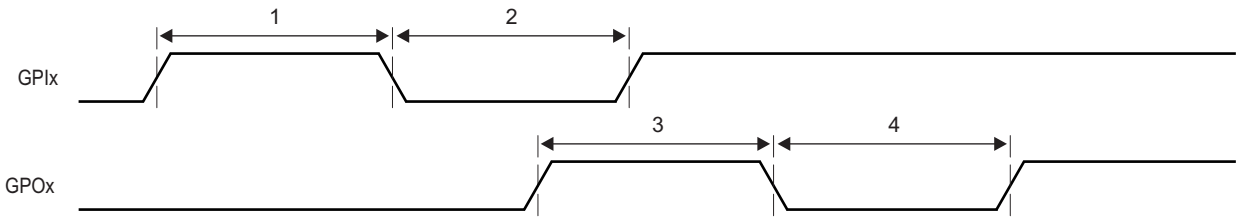


Figure 10-48. GPIO Timing

10.21 Semaphore2

The device contains an enhanced Semaphore module for the management of shared resources of the SoC. The Semaphore enforces atomic accesses to shared chip-level resources so that the read-modify-write sequence is not broken. The Semaphore module has unique interrupts to each of the CorePacs to identify when that CorePac has acquired the resource.

Semaphore resources within the module are not tied to specific hardware resources. It is a software requirement to allocate semaphore resources to the hardware resource(s) to be arbitrated.

The Semaphore module supports three masters and contains 64 semaphores that can be shared within the system.

There are two methods of accessing a semaphore resource:

- **Direct Access:** A CorePac directly accesses a semaphore resource. If free, the semaphore is granted. If not free, the semaphore is not granted.
- **Indirect Access:** A CorePac indirectly accesses a semaphore resource by writing to it. Once the resource is free, an interrupt notifies the CorePac that the resource is available.

10.22 Universal Serial Bus 3.0 (USB 3.0)

The device includes a USB 3.0 controller providing the following capabilities:

- Support of USB 3.0 peripheral (or device) mode at the following speeds:
  - Super Speed (SS) (5 Gbps)
  - High Speed (HS) (480 Mbps)
  - Full Speed (FS) (12 Mbps)
- Support of USB 3.0 host mode at the following speeds:
  - Super Speed (SS) (5 Gbps)
  - High Speed (HS) (480 Mbps)
  - Full Speed (FS) (12 Mbps)
  - Low Speed (LS) (1.5 Mbps)
- Integrated DMA controller with extensible Host Controller Interface (xHCI) support
- Support for 14 transmit and 14 receive endpoints plus control EP0

For more information, see the *KeyStone II Architecture Universal Serial Bus 3.0 (USB 3.0) User's Guide* (SPRUHJ7).

10.23 TSIP Peripheral

The Telecom Serial Interface Port (TSIP) module provides a glueless interface to common telecom serial data streams. For more information, see the *KeyStone Architecture Telecom Serial Interface Port (TSIP) User Guide* (SPRUGY4).

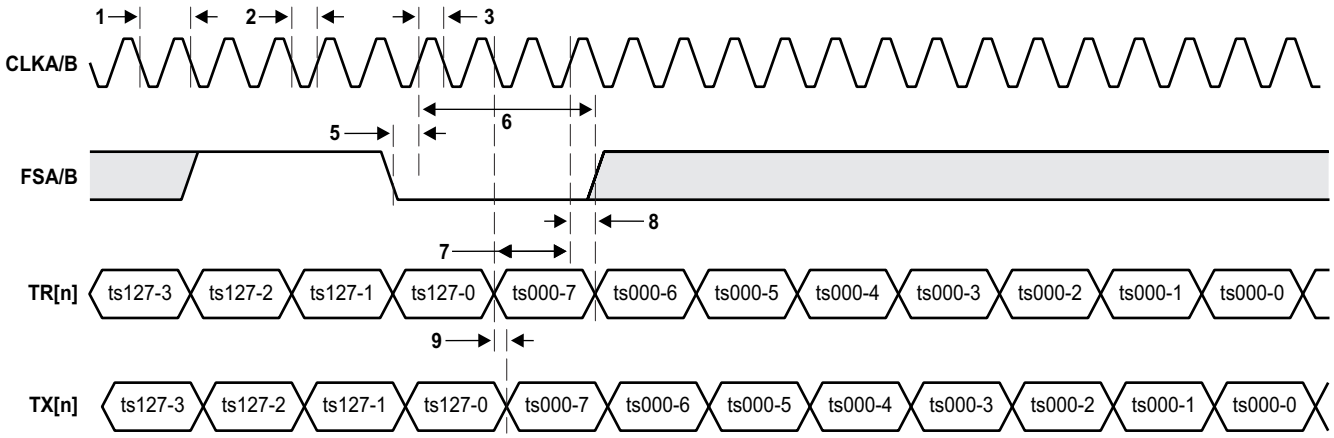
10.23.1 TSIP Electrical Data/Timing

Table 10-54. Timing Requirements for TSIP 2x Mode(1)

(see Figure 10-49)

NO.			MIN	MAX	UNIT
1	t_c(CLK)	Cycle time, CLK rising edge to next CLK rising edge	61(2)		ns
2	t_w(CLKL)	Pulse duration, CLK low	0.4×t_c(CLK)		ns
3	t_w(CLKH)	Pulse duration, CLK high	0.4×t_c(CLK)		ns
4	t_t(CLK)	Transition time, CLK high to low or CLK low to high		2	ns
5	t_{su}(FS-CLK)	Setup time, FS valid before rising CLK	5		ns
6	t_h(CLK-FS)	Hold time, FS valid after rising CLK	5		ns
7	t_{su}(TR-CLK)	Setup time, TR valid before rising CLK	5		ns
8	t_h(CLK-TR)	Hold time, TR valid after rising CLK	5		ns
9	t_d(CLKL-TX)	Delay time, CLK low to TX valid	1	12	ns
10	t_{dis}(CLKH-TXZ)	Disable time, CLK low to TX Hi-Z	2	10	ns

(1) Polarities of XMTFSYNCP = 0b, XMTFCLKP = 0, XMTDCLKP = 1b, RCVFSYNCP = 0, RCVFCLKP = 0, RCVDCLKP = 0. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
(2) Timing shown is for 8.192 Mbps links. Timing for 16.384 Mbps and 32.768 Mbps links is 30.5 ns and 15.2 ns, respectively.



A. Example timeslot numbering shown is for 8.192 Mbps links; 16.384 Mbps links have timeslots numbered 0 through 255 and 32.768 Mbps links have timeslots numbered 0 through 511. The data timing shown relative to the clock and frame sync signals would require a RCVDATD=1 and a XMTDATD=1

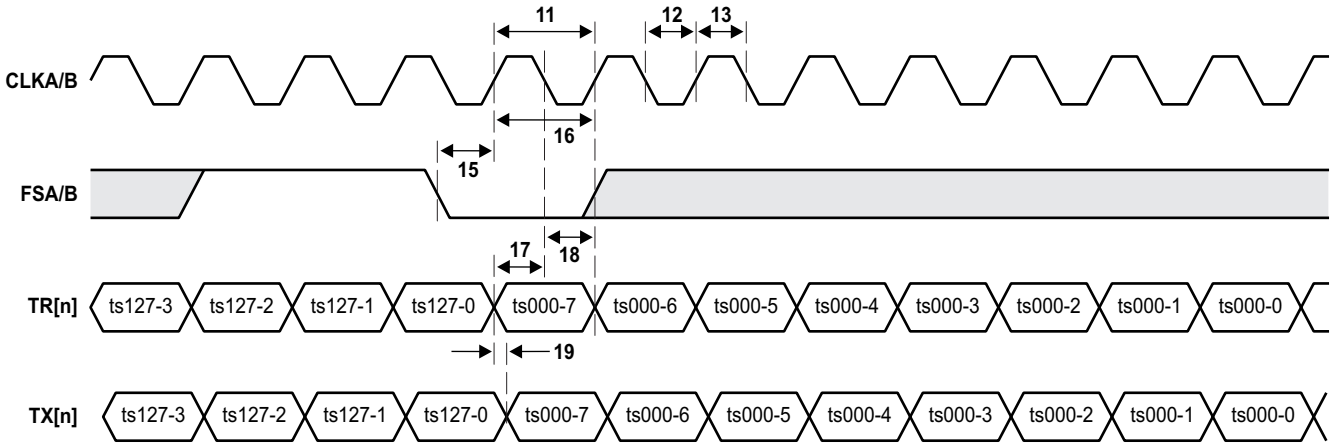
Figure 10-49. TSIP 2x Timing Diagram(A)

Table 10-55. Timing Requirements for TSIP 1x Mode(1)

(see Figure 10-50)

NO.			MIN	MAX	UNIT
11	t_c(CLK)	Cycle time, CLK rising edge to next CLK rising edge	122.1(2)		ns
12	t_w(CLKL)	Pulse duration, CLK low	0.4×t_c(CLK)		ns
13	t_w(CLKH)	Pulse duration, CLK high	0.4×t_c(CLK)		ns
14	t_t(CLK)	Transition time, CLK high to low or CLK low to high		2	ns
15	t_{su}(FS-CLK)	Setup time, FS valid before rising CLK	5		ns
16	t_h(CLK-FS)	Hold time, FS valid after rising CLK	5		ns
17	t_{su}(TR-CLK)	Setup time, TR valid before rising CLK	5		ns
18	t_h(CLK-TR)	Hold time, TR valid after rising CLK	5		ns
19	t_d(CLKL-TX)	Delay time, CLK low to TX valid	1	12	ns
20	t_{dis}(CLKH-TXZ)	Disable time, CLK low to TX Hi-Z	2	10	ns

(1) Polarities of XMTFSYNCP = 0b, XMTFCLKP = 0, XMTDCLKP = 0b, RCVFSYNCP = 0, RCVFCLKP = 0, RCVDCLKP = 1. If the polarity of any of the signals is inverted, then the timing references of that signal are also inverted.
(2) Timing shown is for 8.192 Mbps links. Timing for 16.384 Mbps and 32.768 Mbps links is 61 ns and 30.5 ns, respectively.



A. Example timeslot numbering shown is for 8.192 Mbps links; 16.384 Mbps links have timeslots numbered 0 through 255 and 32.768 Mbps links have timeslots numbered 0 through 511. The data timing shown relative to the clock and frame sync signals would require a RCVDATD=1023 and a XMTDATD=1023.

Figure 10-50. TSIP 1x Timing Diagram(A)

10.24 Universal Subscriber Identity Module (USIM)

The AM5K2E0x is equipped with a Universal Subscriber Identity Module (USIM) for user authentication. The USIM is compatible with ISO, ETSI/GSM, and 3GPP standards.

The USIM is implemented for support of secure devices only. Contact your local technical sales representative for further details.

10.25 EMIF16 Peripheral

The EMIF16 module provides an interface between the device and external memories such as NAND and NOR flash. For more information, see the *KeyStone Architecture External Memory Interface (EMIF16) User's Guide* (SPRUGZ3).

10.25.1 EMIF16 Electrical Data/Timing

Table 10-56. EMIF16 Asynchronous Memory Timing Requirements(1)

(see Figure 10-51 through Figure 10-54)

NO.			MIN	MAX	UNIT
General Timing					
2	t_w(WAIT)	Pulse duration, WAIT assertion and deassertion minimum time		2E	ns
28	t_d(WAIT-WEH)	Setup time, WAIT asserted before WE high		4E + 3	ns
14	t_d(WAIT-OEH)	Setup time, WAIT asserted before OE high		4E + 3	ns
Read Timing					
3	t_C(CEL)	EMIF read cycle time when ew = 0, meaning not in extended wait mode	(RS+RST+RH+3) *E-3	(RS+RST+RH+3) *E+3	ns
3	t_C(CEL)	EMIF read cycle time when ew =1, meaning extended wait mode enabled	(RS+RST+RH+3) *E-3	(RS+RST+RH+3) *E+3	ns
4	t_{osu}(CEL-OEL)	Output setup time from CE low to OE low. SS = 0, not in select strobe mode	(RS+1) * E - 3	(RS+1) * E + 3	ns
5	t_{oh}(OEH-CEH)	Output hold time from OE high to CE high. SS = 0, not in select strobe mode	(RH+1) * E - 3	(RH+1) * E + 3	ns
4	t_{osu}(CEL-OEL)	Output setup time from CE low to OE low in select strobe mode, SS = 1	(RS+1) * E - 3	(RS+1) * E + 3	ns
5	t_{oh}(OEH-CEH)	Output hold time from OE high to CE high in select strobe mode, SS = 1	(RH+1) * E - 3	(RH+1) * E + 3	ns
6	t_{osu}(BAV-OEL)	Output setup time from BA valid to OE low	(RS+1) * E - 3	(RS+1) * E + 3	ns

(1) E = 1/(SYSCLK1/6)

Table 10-56. EMIF16 Asynchronous Memory Timing Requirements[(1)] (continued)

(see Figure 10-51 through Figure 10-54)

NO.			MIN	MAX	UNIT
7	t_{oh}(OEH-BAIV)	Output hold time from OE high to BA invalid	(RH+1) * E - 3	(RH+1) * E + 3	ns
8	t_{osu}(AV-OEL)	Output setup time from A valid to OE low	(RS+1) * E - 3	(RS+1) * E + 3	ns
9	t_{oh}(OEH-AIV)	Output hold time from OE high to A invalid	(RH+1) * E - 3	(RH+1) * E + 3	ns
10	t_w(OEL)	OE active time low, when ew = 0. Extended wait mode is disabled.	(RST+1) * E - 3	(RST+1) * E + 3	ns
10	t_w(OEL)	OE active time low, when ew = 1. Extended wait mode is enabled.	(RST+1) * E - 3	(RST+1) * E + 3	ns
11	t_d(WAITH-OEH)	Delay time from WAIT deasserted to OE# high		4E + 3	ns
12	t_{su}(D-OEH)	Input setup time from D valid to OE high	3		ns
13	t_h(OEH-D)	Input hold time from OE high to D invalid	0.5		ns
Write Timing					
15	t_c(CEL)	EMIF write cycle time when ew = 0, meaning not in extended wait mode	(WS+WST+WH+3)*E-3	(WS+WST+WH+3)*E+3	ns
15	t_c(CEL)	EMIF write cycle time when ew =1., meaning extended wait mode is enabled	(WS+WST+WH+3)*E-3	(WS+WST+WH+3)*E+3	ns
16	t_{osu}CEL-WEL)	Output setup time from CE low to WE low. SS = 0, not in select strobe mode	(WS+1) * E - 3		ns
17	t_{oh}(WEH-CEH)	Output hold time from WE high to CE high. SS = 0, not in select strobe mode	(WH+1) * E - 3		ns
16	t_{osu}CEL-WEL)	Output setup time from CE low to WE low in select strobe mode, SS = 1	(WS+1) * E - 3		ns
17	t_{oh}(WEH-CEH)	Output hold time from WE high to CE high in select strobe mode, SS = 1	(WH+1) * E - 3		ns
18	t_{osu}(RNW-WEL)	Output setup time from RNW valid to WE low	(WS+1) * E - 3		ns
19	t_{oh}(WEH-RNW)	Output hold time from WE high to RNW invalid	(WH+1) * E - 3		ns
20	t_{osu}(BAV-WEL)	Output setup time from BA valid to WE low	(WS+1) * E - 3		ns
21	t_{oh}(WEH-BAIV)	Output hold time from WE high to BA invalid	(WH+1) * E - 3		ns
22	t_{osu}(AV-WEL)	Output setup time from A valid to WE low	(WS+1) * E - 3		ns
23	t_{oh}(WEH-AIV)	Output hold time from WE high to A invalid	(WH+1) * E - 3		ns
24	t_w(WEL)	WE active time low, when ew = 0. Extended wait mode is disabled.	(WST+1) * E - 3		ns
24	t_w(WEL)	WE active time low, when ew = 1. Extended wait mode is enabled.	(WST+1) * E - 3		ns
26	t_{osu}(DV-WEL)	Output setup time from D valid to WE low	(WS+1) * E - 3		ns
27	t_{oh}(WEH-DIV)	Output hold time from WE high to D invalid	(WH+1) * E - 3		ns
25	t_d(WAITH-WEH)	Delay time from WAIT deasserted to WE# high		4E + 3	ns

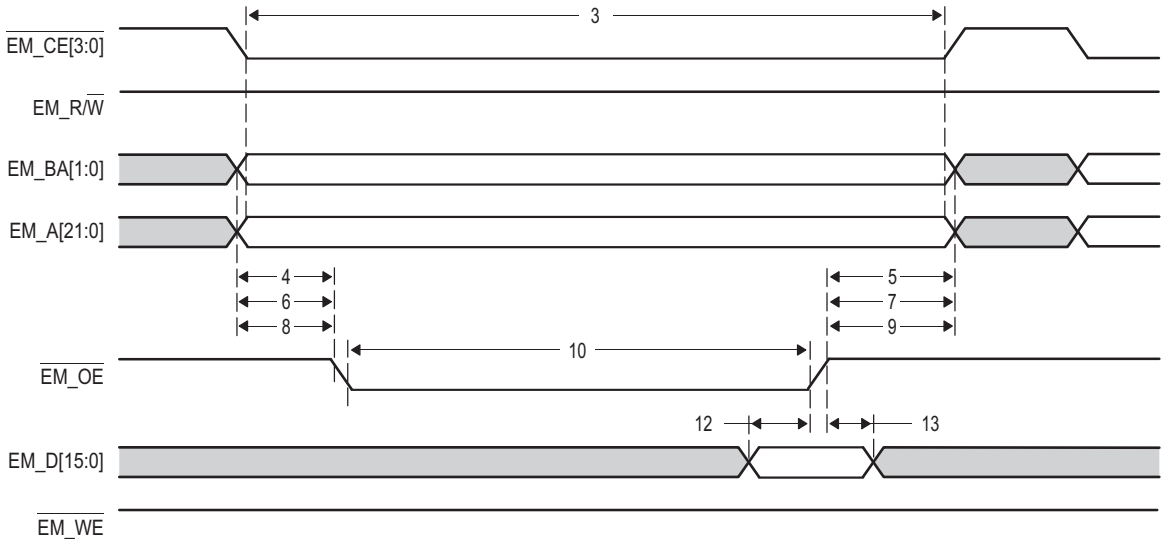


Figure 10-51. EMIF16 Asynchronous Memory Read Timing Diagram

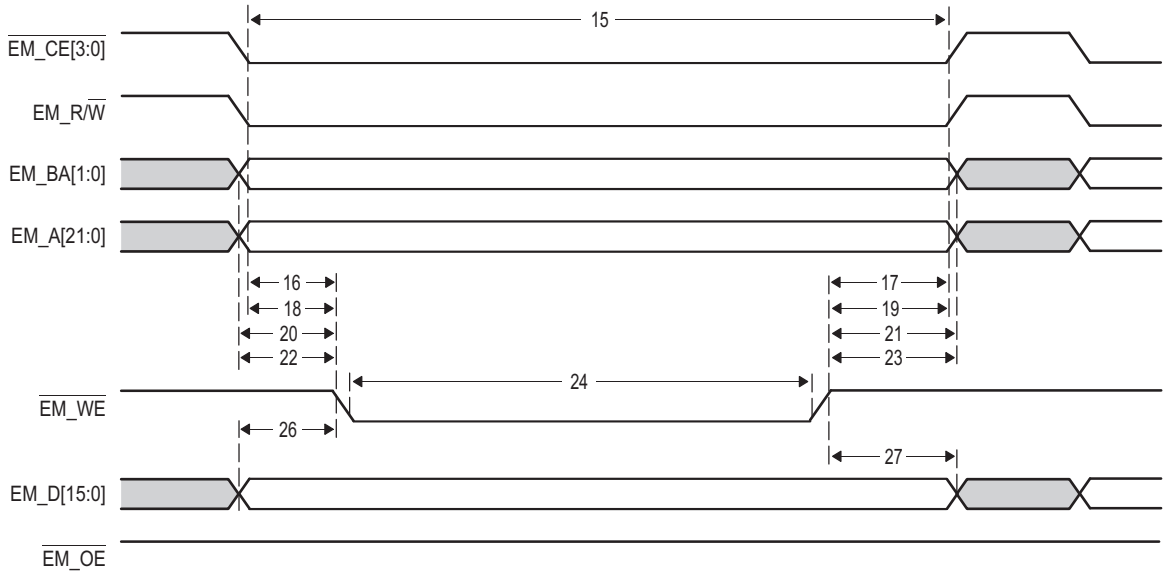


Figure 10-52. EMIF16 Asynchronous Memory Write Timing Diagram

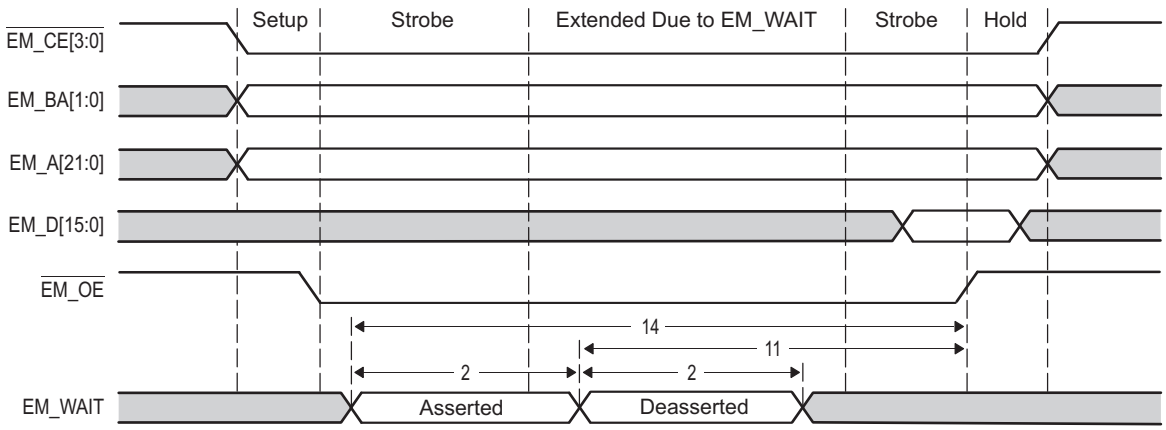


Figure 10-53. EMIF16 EM_WAIT Read Timing Diagram

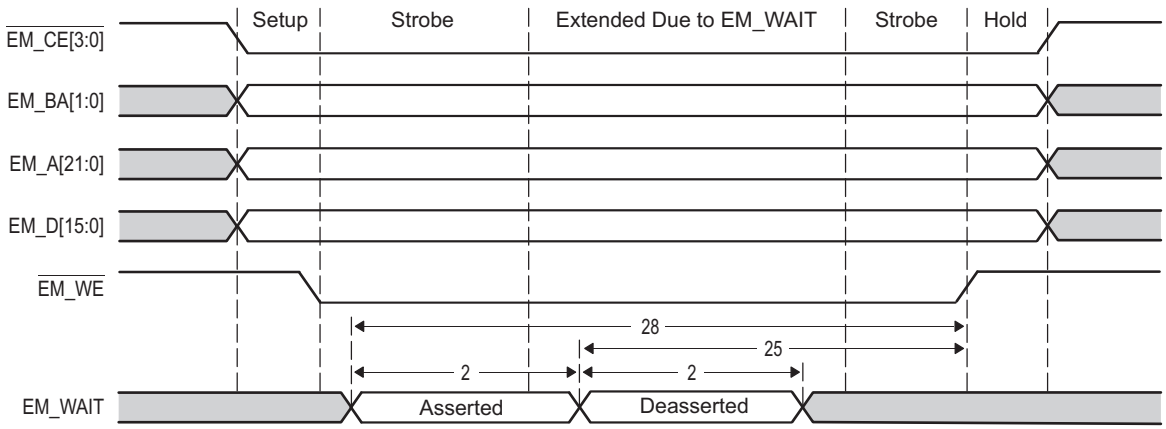


Figure 10-54. EMIF16 EM_WAIT Write Timing Diagram

10.26 Emulation Features and Capability

The debug capabilities of KeyStone II devices include the Debug subsystem module (DEBUGSS). The DEBUGSS module contains the ICEPick module which handles the external JTAG Test Access Port (TAP) and multiple secondary TAPs for the various processing cores of the device. It also provides Debug Access Port (DAP) for system wide memory access from debugger, Cross triggering, System trace, Peripheral suspend generation, Debug port (EMUx) pin management etc. The DEBUGSS module works in conjunction with the debug capability integrated in the processing cores to provide a comprehensive hardware platform for a rich debug and development experience.

10.26.1 Chip Level Features

- Support for 1149.1(JTAG and Boundary scan) and 1149.6 (Boundary scan extensions).
- Trace sources to DEBUG SubSystem System Trace Module (DEBUGSS STM)
  - Provides a way for hardware instrumentation and software messaging to supplement the processor core trace mechanisms.
  - Hardware instrumentation support of CPTracers to support logging of bus transactions for critical endpoints
  - Software messaging/instrumentation support for SoC and QMSS PDSP cores through DEBUGSS STM.
- Trace Sinks
  - Support for trace export (from all processor cores and DEBUGSS STM) through emulation pins. Concurrent trace of ARM and STM traces via EMU pins is possible.
  - Support for 32KB DEBUGSS TBR (Trace Buffer and Router) to hold system trace. The data can be drained using EDMA to on-chip or DDR memory buffers. These intermediate buffers can subsequently be drained through the device high speed interfaces. The DEBUGSS TBR is dedicated to the DEBUGSS STM module. The trace draining interface used in KeyStone II for DEBUGSS and ARMSS are based on the new CT-TBR.
- Cross triggering: Provides a way to propagate debug (trigger) events from one processor/subsystem/module to another
  - Cross triggering between multiple devices via EMU0/EMU1 pins
  - Cross triggering between multiple processing cores within the device like ARM Cores and non-processor entities like ARM STM (input only), CPTracers, CT-TBRs and DEBUGSS STM (input only)
- Synchronized starting and stopping of processing cores
  - Global start of all ARM cores
  - Global stopping of all ARM cores
- Emulation mode aware peripherals (suspend features and debug access features)
- Support system memory access via the DAP port (natively support 32-bit address, and it can support 36-bit address through configuration of MPAX inside MSMC). Debug access to any invalid memory location (reserved/clock-gated/power-down) does not cause system hang.
- Scan access to secondary TAPs of DEBUGSS is disabled in Secure devices by default. Security override sequence is supported (requires software override sequence) to enable debug in secure devices. In addition, Debug features of the ARM cores are blockable through the ARM debug authentication interface in secure devices.
- Support WIR (wait-in-reset) debug boot mode for Non-secure devices.
- Debug functionality survives all pin resets except power-on resets ($\overline{\text{POR}}$/$\overline{\text{RESETFULL}}$) and test reset ($\overline{\text{TRST}}$).
- PDSP Debug features like access/control through DAP, Halt mode debug and software instrumentation.

10.26.1.1 ARM Subsystem Features

- Support for invasive debug like halt mode debugging (breakpoint, watchpoints) and monitor mode debugging

- Support for non-invasive debugging (program trace, performance monitoring)
- Support for A15 Performance Monitoring Unit (cycle counters)
- Support for per core CoreSight™ Program Trace Module (CS-PTM) with timing
- Support for an integrated CoreSight System Trace Module (CS-STM) for hardware event and software instrumentation
- A shared timestamp counter for all ARM cores and STM is integrated in ARMSS for trace data correlation
- Support for a 16KB Trace Buffer and Router (TBR) to hold PTM/STM trace. The trace data is copied by EDMA to external memory for draining by device high speed serial interfaces.
- Support for simultaneous draining of trace stream through EMUn pins and TBR (to achieve higher aggregate trace throughput)
- Support for debug authentication interface to disable debug accesses in secure devices
- Support for cross triggering between MPU cores, CS-STM and CT-TBR
- Support for debug through warm reset

10.26.2 ICEPick Module

The debugger is connected to the device through its external JTAG interface. The first level of debug interface seen by the debugger is connected to the ICEPick module embedded in the DEBUGSS. ICEPick is the chip-level TAP, responsible for providing access to the IEEE 1149.1 and IEEE1149.6 boundary scan capabilities of the device.

ICEPick manages the TAPs as well as the power/reset/clock controls for the logic associated with the TAPs as well as the logic associated with the APB ports.

ICEPick provides the following debug capabilities:

- Debug connect logic for enabling or disabling most ICEPick instructions
- Dynamic TAP insertion
  - Serially linking up to 32 TAP controllers
  - Individually selecting one or more of the TAPS for scan without disrupting the instruction register (IR) state of other TAPs
- Power, reset and clock management
  - Provides the power and clock status of the domain to the debugger
  - Provides debugger control of the power domain of a processor.
    - Force the domain power and clocks on
    - Prohibit the domain from being clock-gated or powered down
  - Applies system reset
  - Provides wait-in-reset (WIR) boot mode
  - Provides global and local WIR release
  - Provides global and local reset block

The ICEPick module implements a connect register, which must be configured with a predefined key to enable the full set of JTAG instructions. Once the debug connect key has been properly programmed, ICEPick signals and subsystems emulation logic should be turned on.

10.26.2.1 ICEPick Dynamic Tap Insertion

To include more or fewer secondary TAPS in the scan chain, the debugger must use the ICEPick TAP router to program the TAPs. At its root, ICEPick is a scan-path linker that lets the debugger selectively choose which subsystem TAPs are accessible through the device-level debug interface. Each secondary TAP can be dynamically included in or excluded from the scan path. From external JTAG interface point of view, secondary TAPS that are not selected appear not to exist.

The CoreSight components are interfaced with ICEPick through the CS_DAP module. The CS_DAP is attached to the ICEPick secondary TAP and translates JTAG transactions into APBv3 transactions.

Table 10-57 shows the ICEPick secondary taps in the system. For more details on the test related P1500 TAPs, see the DFTSS specification.

Table 10-57. ICEPick Debug Secondary TAPs

TAP #	TYPE	NAME	IR SCAN LENGTH	ACCESS IN SECURE DEVICE	DESCRIPTION
0	n/a	n/a	n/a	No	Reserved (This is an internal TAP and not exposed at the DEBUGSS boundary)
1	JTAG				Reserved
2	JTAG				Reserved
3	JTAG				Reserved
4	JTAG				Reserved
5	JTAG				Reserved
6	JTAG				Reserved
7	JTAG				Reserved
8	JTAG				Reserved
9..13	JTAG	Reserved	NA	No	Spare ports for future expansion
14	CS	CS_DAP (APB-AP)	4	No	ARM A15 Cores (This is an internal TAP and not exposed at the DEBUGSS boundary)
		CS_DAP (AHB-AP)			PDSP Cores (This is an internal TAP and not exposed at the DEBUGSS boundary)

For more information on ICEPick, see the *KeyStone II Architecture Debug and Trace User's Guide* (SPRUHM4).

10.27 Debug Port (EMUx)

The device also supports 34 emulation pins — EMU[33:0], which includes 19 dedicated EMU pins and 15 pins multiplexed with GPIO. These pins are shared by SoC STM trace, cross triggering, and debug boot modes as shown in Table 10-60. The 34-pin dedicated emulation interface is also defined in the following table.

NOTE

Note that if EMU[1:0] signals are shared for cross-triggering purposes in the board level, they SHOULD NOT be used for trace purposes.

10.27.1 Concurrent Use of Debug Port

The following combinations are possible concurrently:

- Trigger 0/1
- Trigger 0/1 and STM Trace (up to 4 data pins)
- Trigger 0/1 and STM Trace (up to 4 data pins)
- Trigger 0/1 and STM Trace (1-4 data pins) and ARM Trace (27-24 data pins)
- STM Trace (1-4 data pins) and ARM Trace (29-26 data pins)
- Trigger 0/1 and ARM Trace (up to 29 data pins)
- ARM Trace (up to 32 data pins)

10.27.2 Master ID for HW and SW Messages

Table 10-58 describes the master ID for the various hardware and software masters of the STM.

Table 10-58. MSTID Mapping for Hardware Instrumentation (CPTRACERS)

CPTRACER NAME	MSTID [7:0]	CLOCK DOMAIN	SID[4:0]	DESCRIPTION
CPT_MSMCx_MST, where x = 0..3	0x94-0x97	SYSCLK1/1	0x0..3	MSMC SRAM Bank 0 to MSMC SRAM Bank 3 monitors
CPT_MSMC4_MST	0xB1	SYSCLK1/1	0x4	MSMC SRAM Bank 4
CPT_MSMCx_MST, where x = 5..7	0xAE - 0xB0	SYSCLK1/1	0x5..7	MSMC SRAM Bank 5to MSMC SRAM Bank 7 monitors
CPT_DDR3_MST	0x98	SYSCLK1/1	0x8	MSMC DDR3 port monitor
CPT_L2_x_MST, where x = 0..7	0x8C - 0x93	SYSCLK1/3	0x9..0x10	Reserved
CPT_TPCC0_4_MST	0xA4	SYSCLK1/3	0x11	EDMA 0 and EDMA 4 CFG port monitor
CPT_TPCC1_2_3_MST	0xA5	SYSCLK1/3	0x12	EDMA 1, EDMA2 and EDMA3 CFG port monitor
CPT_INTC_MST	0xA6	SYSCLK1/3	0x13	INTC port monitor (for INTC 0/1/2 and GIC400)
CPT_SM_MST	0x99	SYSCLK1/3	0x14	Semaphore CFG port monitors
CPT_QM_CFG1_MST	0x9A	SYSCLK1/3	0x15	QMSS CFG1 port monitor
CPT_QM_CFG2_MST	0xA0	SYSCLK1/3	0x16	QMSS CFG2 port monitor
CPT_QM_M_MST	0x9B	SYSCLK1/3	0x17	QM_M CFG/DMA port monitor
CPT_SPI_ROM_EMIF16_MST	0xA7	SYSCLK1/3	0x18	SPI ROM EMIF16 CFG port monitor
CPT_CFG_MST	0x9C	SYSCLK1/3	0x19	SCR_3P_B and SCR_6P_B CFG peripheral port monitors
Reserved			0x1A	Reserved
Reserved			0x1B	Reserved
Reserved			0x1C	Reserved
Reserved			0x1D	Reserved
Reserved			0x1E	Reserved
Reserved			0x1F	DDR 3B port monitor (on SCR 3C)

Table 10-59. MSTID Mapping for Software Messages

CORE NAME	MSTID [7:0]	DESCRIPTION
Reserved	0x0	
Reserved	0x1	
Reserved	0x2	
Reserved	0x3	
Reserved	0x4	
Reserved	0x5	
Reserved	0x6	

Table 10-59. MSTID Mapping for Software Messages (continued)

CORE NAME	MSTID [7:0]	DESCRIPTION
Reserved	0x7	
A15 Core0	0x8	ARM Master IDs
A15 Core1	0x9	ARM Master ID (AM5K2E04 only)
A15 Core2	0xA	ARM Master ID(AM5K2E04 only)
A15 Core3	0xB	ARM Master ID(AM5K2E04 only)
QMSS PDSPs	0x46	All QMSS PDSPs share the same master ID. Differentiating between the 8 PDSPs is done through the channel number used
TSIP	0x80	TSIP Master ID

10.27.3 SoC Cross-Triggering Connection

The cross-trigger lines are shared by all the subsystems implementing cross-triggering. An MPU subsystem trigger event can therefore be propagated to any application subsystem or system trace component. The remote subsystem or system trace component can be programmed to be sensitive to the global SOC trigger lines to either:

- Generate a processor debug request
- Generate an interrupt request
- Start/Stop processor trace
- Start/Stop CBA transaction tracing through CPTracers
- Start external logic analyzer trace
- Stop external logic analyzer trace

Table 10-60. Cross-Triggering Connection

NAME	SOURCE TRIGGERS	SINK TRIGGERS	COMMENTS
Inside DEBUGSS			
Device-to-device trigger via EMU0/1 pins	YES	YES	This is fixed (not affected by configuration)
MIPI-STM	NO	YES	Trigger input only for MIPI-STM in DebugSS
CT-TBR	YES	YES	DEBUGSS CT-TBR
CS-TPIU	NO	YES	DEBUGSS CS-TPIU
Outside DEBUGSS			
CP_Tracers	YES	YES	
ARM	YES	YES	ARM Cores, ARM CS-STM and ARM CT-TBR

The following table describes the crosstrigger connection between various cross trigger sources and TI XTRIG module.

Table 10-61. TI XTRIG Assignment

NAME	ASSIGNED XTRIG CHANNEL NUMBER
CPTracer 0..31 (The CPTracer number refers to the SID[4:0] as shown in Table 10-58	XTRIG 8 .. 39

10.27.4 Peripherals-Related Debug Requirement

Table 10-62 lists all the peripherals on this device, and the status of whether or not it supports emulation suspend or emulation request events.

The DEBUGSS supports upto 32 debug suspend sources (processor cores) and 64 debug suspend sinks (peripherals). The assignment of processor cores is shown in and the assignment of peripherals is shown in Table 10-62. By default the logical AND of all the processor cores is routed to the peripherals. It is possible to select an individual core to be routed to the peripheral (For example: used in tightly coupled peripherals like timers), a logical AND of all cores (Global peripherals) or a logical OR of all cores by programming the DEBUGSS.DRM module.

The SOFT bit should be programmed based on whether or not an immediate pause of the peripheral function is required or if the peripheral suspend should occur only after a particular completion point is reached in the normal peripheral operation. The FREE bit should be programmed to enable or disable the emulation suspend functionality.

Table 10-62. Peripherals Emulation Support

PERIPHERAL	EMULATION SUSPEND SUPPORT				EMULATION REQUEST SUPPORT (cemudbg/emudbg)	DEBUG PERIPHERAL ASSIGNMENT
	STOP-MODE	REAL-TIME MODE	FREE BIT	STOP BIT		
Infrastructure Peripherals						
EDMA_x, where X=0/1/2/3/4	N	N	N	N	Y	NA
QM_SS	Y (CPDMA only)	Y (CPDMA only)	Y (CPDMA only)	Y (CPDMA only)	Y	20
CP_Tracers_X, where X = 0..32	N	N	N	N	N	NA
MPU_X, where X = 0..11	N	N	N	N	Y	NA
CP_INTC	N	N	N	N	Y	NA
BOOT_CFG	N	N	N	N	Y	NA
SEC_MGR	N	N	N	N	Y	NA
PSC	N	N	N	N	N	NA
PLL	N	N	N	N	N	NA
TIMERx, x=0, 1..7, 8..19	Y	N	Y	Y	N	0, 1..7, 8..19
Semaphore	N	N	N	N	Y	NA
GPIO	N	N	N	N	N	NA
Memory Controller Peripherals						
DDR3	N	N	N	N	Y	NA
MSMC	N	N	N	N	Y	NA
EMIF16	N	N	N	N	Y	NA
Serial Interfaces						
I^2C_X, where X = 0/1/2	Y	N	Y	Y	Y	21/22/23
SPI_X, where X = 0/1/2	N	N	N	N	Y	NA
UART_X, where X = 0/1	Y	N	Y	Y	Y	24/25
USIM	Y	N	Y	N	N	28
High Speed Serial Interfaces						
Hyperlink	N	N	N	N	Y	
PCIeSS 0..1	N	N	N	N	N	
Reserved						26
NetCP (ethernet switch)	Y	Y	Y	Y	N	27

Table 10-62. Peripherals Emulation Support (continued)

PERIPHERAL	EMULATION SUSPEND SUPPORT				EMULATION REQUEST SUPPORT (cemudbg/emudbg)	DEBUG PERIPHERAL ASSIGNMENT
	STOP-MODE	REAL-TIME MODE	FREE BIT	STOP BIT		
10GbE (ethernet switch)(1)	Y	N	Y	Y	N	29
USBSS	N	N	N	N	N	NA

(1) 10 GbE supported by AM5K2E04 only.

Based on the above table the number of suspend interfaces in Keystone II devices is listed below.

Table 10-63. EMUSUSP Peripheral Summary (for EMUSUSP handshake from DEBUGSS)

INTERFACES	NUM_SUSPEND_PERIPHERALS
EMUSUSP Interfaces	54
EMUSUSP Realtime Interfaces	15

Table 10-64 summarizes the DEBUG core assignment. Emulation suspend output of all the cores are synchronized to SYSCLK1/6 which is frequency of the slowest peripheral that uses these signals.

Table 10-64. EMUSUSP Core Summary(for EMUSUSP handshake to DEBUGSS)

Core #	Assignment
8..11	ARM CorePac0-3
12..29	Reserved
30	Logical OR of Core #0..11
31	Logical AND of Core #0..11

10.27.5 Advanced Event Triggering (AET)

The device supports advanced event triggering (AET). This capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- **Hardware program breakpoints**: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- **Data watchpoints**: specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- **Counters**: count the occurrence of an event or cycles for performance monitoring.
- **State sequencing**: allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

For more information on the AET, see the following documents:

- *Using Advanced Event Triggering to Find and Fix Intermittent Real-Time Bugs* application report (SPRA753)
- *Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor Systems* application report (SPRA387)

10.27.6 Trace

The device supports trace. Trace is a debug technology that provides a detailed, historical account of application code execution, timing, and data accesses. Trace collects, compresses, and exports debug information for analysis. Trace works in real-time and does not impact the execution of the system.

For more information on board design guidelines for trace advanced emulation, see the *Emulation and Trace Headers Technical Reference Manual* (SPRU655).

10.27.6.1 Trace Electrical Data/Timing

Table 10-65. Trace Switching Characteristics

(see Figure 10-55)

NO.	PARAMETER		MIN	MAX	UNIT
1	t_w(DPnH)	Pulse duration, DPn/EMUn high	2.4		ns
1	t_w(DPnH)90%	Pulse duration, DPn/EMUn high detected at 90% Voh	1.5		ns
2	t_w(DPnL)	Pulse duration, DPn/EMUn low	2.4		ns
2	t_w(DPnL)10%	Pulse duration, DPn/EMUn low detected at 10% Voh	1.5		ns
3	t_{sko}(DPn)	Output skew time, time delay difference between DPn/EMUn pins configured as trace	-1	1	ns
	t_{skp}(DPn)	Pulse skew, magnitude of difference between high-to-low (tphl) and low-to-high (tplh) propagation delays.		600	ps
	t_{sldp_o}(DPn)	Output slew rate DPn/EMUn	3.3		V/ns

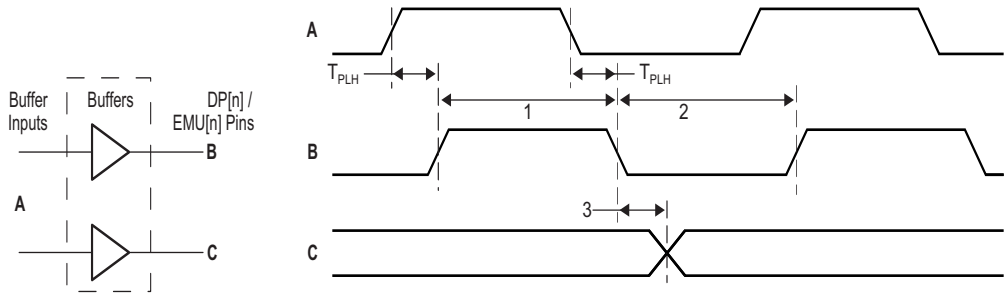


Figure 10-55. Trace Timing

10.27.7 IEEE 1149.1 JTAG

The Joint Test Action Group (JTAG) interface is used to support boundary scan and emulation of the device. The boundary scan supported allows for an asynchronous test reset ($\overline{\text{TRST}}$) and only the five baseline JTAG signals (e.g., no EMU[1:0]) required for boundary scan. Most interfaces on the device follow the Boundary Scan Test Specification (IEEE1149.1), while all of the SerDes (SGMII) support the AC-coupled net test defined in AC-Coupled Net Test Specification (IEEE1149.6).

It is expected that all compliant devices are connected through the same JTAG interface, in daisy-chain fashion, in accordance with the specification. The JTAG interface uses 1.8-V LVCMOS buffers, compliant with the *Power Supply Voltage and Interface Standard for Nonterminated Digital Integrated Circuit Specification* (EAI/JESD8-5).

10.27.7.1 IEEE 1149.1 JTAG Compatibility Statement

For maximum reliability, the AM5K2E0x device includes an internal pulldown (IPD) on the $\overline{\text{TRST}}$ pin to ensure that $\overline{\text{TRST}}$ will always be asserted upon power up and the device's internal emulation logic will always be properly initialized when this pin is not routed out. JTAG controllers from Texas Instruments actively drive $\overline{\text{TRST}}$ high. However, some third-party JTAG controllers may not drive $\overline{\text{TRST}}$ high, but expect the use of an external pullup resistor on $\overline{\text{TRST}}$. When using this type of JTAG controller, assert $\overline{\text{TRST}}$ to initialize the device after powerup and externally drive $\overline{\text{TRST}}$ high before attempting any emulation or boundary scan operations.

10.27.7.2 JTAG Electrical Data/Timing

Table 10-66. JTAG Test Port Timing Requirements

(see Figure 10-56)

NO.			MIN	MAX	UNIT
1	$t_{c(TCK)}$	Cycle time, TCK	23		ns
1a	tw(TCKH)	Pulse duration, TCK high (40% of tc)	9.2		ns
1b	tw(TCKL)	Pulse duration, TCK low(40% of tc)	9.2		ns
3	tsu(TDI-TCK)	Input setup time, TDI valid to TCK high	2		ns
3	tsu(TMS-TCK)	Input setup time, TMS valid to TCK high	2		ns
4	th(TCK-TDI)	Input hold time, TDI valid from TCK high	10		ns
4	th(TCK-TMS)	Input hold time, TMS valid from TCK high	10		ns

Table 10-67. JTAG Test Port Switching Characteristics

(see Figure 10-56)

NO.	PARAMETER		MIN	MAX	UNIT
2	$t_{d(TCKL-TDOV)}$	Delay time, TCK low to TDO valid		8.24	ns

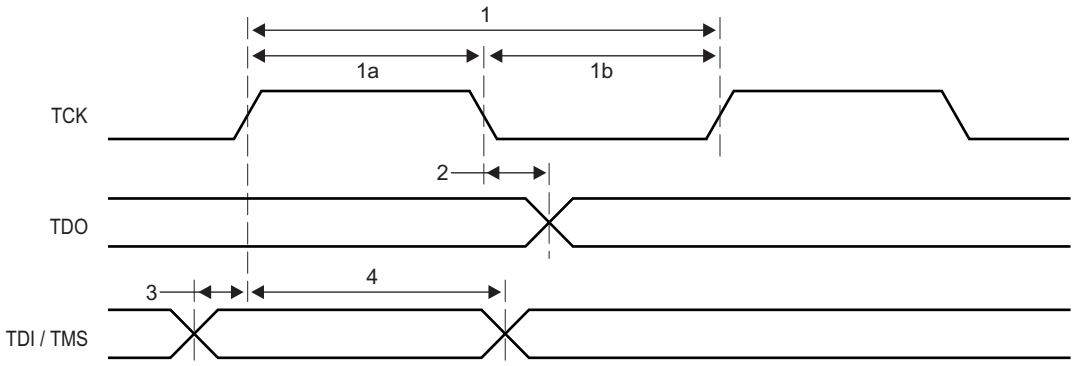


Figure 10-56. JTAG Test-Port Timing

11 Mechanical Data

11.1 Thermal Data

Table 11-1 shows the thermal resistance characteristics for the PBGA - ABD 1089-pin mechanical package.

Table 11-1. Thermal Resistance Characteristics (PBGA Package) ABD

NO.			°C/W
1	$R\theta_{JC}$	Junction-to-case	0.34
2	$R\theta_{JB}$	Junction-to-board	3.14

11.2 Packaging Information

The following packaging information reflects the most current released data available for the designated device(s). This data is subject to change without notice and without revision of this document.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead finish/ Ball material (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
AM5K2E02ABD25	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 85	AM5K2E02ABD @2012 TI	Samples
AM5K2E02ABD4	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 85	AM5K2E02ABD @2012 TI 1.4GHZ	Samples
AM5K2E02ABDA25	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	-40 to 100	AM5K2E02ABD A1.25GHZ	Samples
AM5K2E02ABDA4	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	-40 to 100	AM5K2E02ABD @2012 TI A1.4GHZ	Samples
AM5K2E02XABD25	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 85	AM5K2E02XABD	Samples
AM5K2E04XABD25	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 85	AM5K2E04XABD @2012 TI	Samples
AM5K2E04XABD4	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	0 to 85	AM5K2E04XABD @2012 TI 1.4GHZ	Samples
AM5K2E04XABDA25	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	-40 to 100	AM5K2E04XABD A1.25GHZ	Samples
AM5K2E04XABDA4	ACTIVE	FCBGA	ABD	1089	40	RoHS & Green	Call TI	Level-4-245C-72HR	-40 to 100	AM5K2E04XABD @2012 TI A1.4GHZ	Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

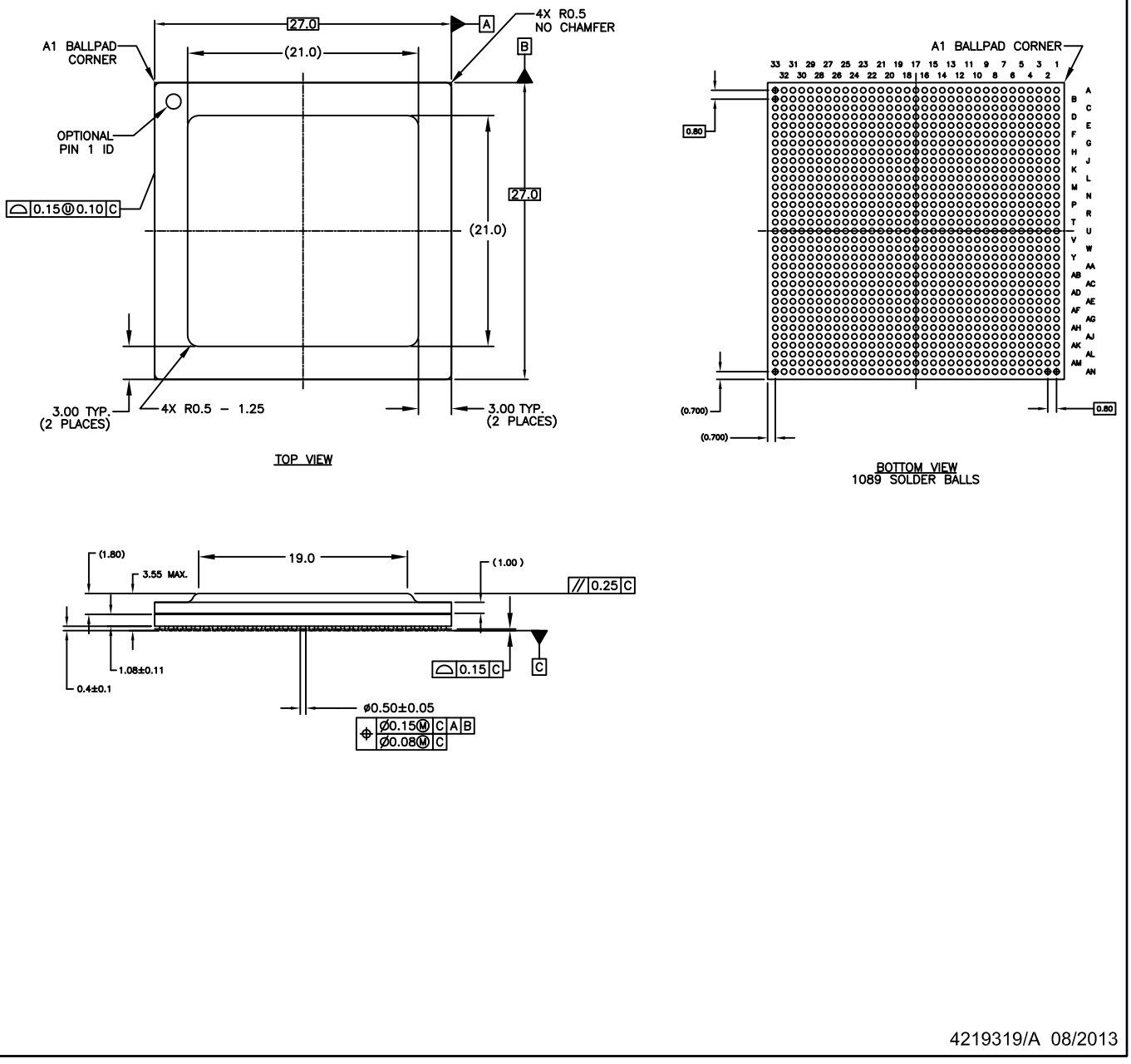
(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead finish/Ball material - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.



4219319/A 08/2013

NOTES:

1. All linear dimensions are in millimeters. Dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.

2. This drawing is subject to change without notice.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265